AD-A082 209 GENERAL SERVICES ADMINISTRATION WASHINGTON DC AUTOMA--ETC F/G 9/2
USE AND SPECIFICATIONS OF REMOTE TERMINAL EMULATION IN ADP SYST--ETC(U)
AUG 79
UNCLASSIFIED FPR-1-4.11 NL

1 OF 3
AD
A082209

FOREWORD

This GSA Handbook, Use and Specifications of Remote Terminal Emulation in ADP System Acquisitions, August 1979, (FPR 1-4.11), is issued by the General Services Administration for use by Federal agencies pursuant to the provisions of Federal Procurement Regulations (FPR) 1-4.11, FPR Temporary Regulation 49, Use of benchmarking and remote terminal emulation for performance validation in the procurement of automated data processing (ADP) systems and services, and Supplement 1 thereto.

The purpose of the handbook is to provide guidance to Federal agencies during ADP system acquisitions to design and to conduct benchmark tests that reflect the current state of technology and that are practical, fair, and equitable for both the Government and ADP vendors.

The handbook has two major objectives. The first is to present significant technical information that will help each Government agency to decide if and how to use remote terminal emulation during an ADP system acquisition. The second objective is to define clearly the range of remote terminal emulation capabilities (1) that an agency is permitted to require offerors to provide for benchmark tests during ADP system acquisitions, and (2) that an offeror must have to be qualified to bid on most Federal ADP system acquisitions.

The handbook reiterates the regulatory limitations that reduce the benchmarking alternatives available to Federal agencies conducting ADP acquisitions. These limitations are intended to protect the interests of the Government by balancing the various acquisition objectives, e.g., assuring competition, meeting specific agency requirements, timeliness, fair dealing, economy, and efficiency. The handbook (1) summarizes introductory concepts and terminology of benchmarking and remote terminal emulation, (2) describes when and how agencies should use remote terminal emulation, and (3) specifies the remote terminal emulation capabilities that an agency may require vendors to provide for testing vendor-proposed ADP systems during acquisitions.

The handbook was prepared by the Federal Computer Performance Evaluation and Simulation Center (FEDSIM) as report NA-018-025-GSA, Use and Specifications of Remote Terminal Emulation in ADP System Acquisitions, dated August 1979. It

A

reflects the comments and suggestions that many Government agencies and ADP vendors offered in response to earlier FEDSIM Working Papers.

Inquiries concerning this handbook may be directed to:

General Services Administration (CDD)
Washington, DC 20405
Telephone Number (202) 566-1076

Limited distribution of Use and Specifications of Remote Terminal Emulation in ADP System Acquisitions, August 1979, (FPR 1-4.11), will be made to agencies submitting requests for copies to General Services Administration/ADTS (CDD) Washington, DC 20405.

Frank J. Carr

FRANK J. CARR
Commissioner

USE AND SPECIFICATIONS OF REMOTE TERMINAL EMULATION IN ADP SYSTEM ACQUISITIONS

GENERAL TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

CHAPTER 1. INTRODUCTION

1. Background.

a. Benchmarking is the process of experimentally imposing a test workload on a set of ADP system components to determine selected execution characteristics of the component(s). Benchmarking is an important part of the competitive ADP acquisition process and can be used to evaluate the capability and capacity of an ADP system or service proposed by a vendor. Remote terminal emulation is one benchmarking technique for conducting tests of teleprocessing (TP) computer systems and services when it is impractical to configure for a test the total planned network of computers, teleprocessing devices, and data communication facilities. Remote terminal emulation uses an external, driver computer and computer programs to imitate the teleprocessing devices to be supported by, and to impose the workload demands on, the actual computer system or service being tested (hereafter referred to as the System Under Test [SUT]). A Remote Terminal Emulator (RTE) is a specific hardware and software implementation of this driver system. During acquisitions, each vendor provides and operates the RTE used for benchmarking that vendor's system. While any benchmark test can be expensive, a benchmark test using remote terminal emulation is usually costly and complex and can be technically invalid if improperly designed or conducted.

b. In 1976, the General Services Administration began a joint Government-industry study of the use of remote terminal emulation in Federal acquisitions of ADP systems and services. The goal of this study was to encourage the use of benchmarking techniques that reflect the current state of technology and are practical, fair, and equitable for both Government and industry. Several factors led to the initiation of this study:

(1) The increasing Government concern for effective and efficient teleprocessing support;

(2) The importance that the Government places on reducing cost risks and mission risks by validating, before contract award, vendor claims of performance;

(3) The increasing use of remote terminal emulation by both Government and industry;

(4) The reduced comparability of benchmark test results caused by a lack of functional similarity between vendors' RTE's;

(5) The possible limiting effects of remote terminal emulation on free and open competition; and

(6) The expense, both to Government and industry, of using remote terminal emulation during acquisition.

c. The Government-industry study, completed in 1979, produced this handbook and a temporary Federal Procurement Regulation (FPR) on the use of benchmarking and remote terminal emulation in Federal ADP acquisitions. The regulation, FPR Temporary Regulation 49 and Supplement 1 thereto, should be studied by all Government and vendor personnel interested in this subject. (The reader should contact GSA to obtain any changes in the regulation and this handbook that have occurred since the issuance of this handbook.) The policy and procedures contained in the regulation for a procurement that acquires an ADP service are different than those for a procurement of an ADP system. An ADP service procurement is defined as an acquisition that results in the Government obtaining the use of ADP equipment (ADPE) containing at least one general purpose, central processing unit (CPU) that is either owned and operated or leased and operated by a contractor. The ADPE may be either dedicated for the exclusive use of the acquiring Government organization or shared by many Government and/or non-Government organizations. An ADP system procurement is defined as an acquisition that results in the lease and/or purchase by the Government of ADPE containing at least one general purpose CPU. The acquired ADPE may be operated by either Government or contractor personnel.

d. The regulation specifies, in summary, that Government agencies shall not require the use of remote terminal emulation in ADP service procurements, except for dedicated requirements or for unusually large, complex, shared requirements. For ADP system procurements, the regulation specifies that each agency shall determine whether or not to require the mandatory use of remote terminal emulation during each of its ADP system procurements. When an agency chooses to use emulation, the agency:

(1) Shall follow all mandatory procedures contained in this handbook;

(2) Shall not require remote terminal emulation capabilities which are not explicitly defined in this handbook;

(3) May declare an offer nonresponsive and may disqualify that offeror from the procurement, if the offeror fails to provide the emulation capabilities required by the solicitation; and

(4) Shall not require an offeror to conduct a benchmark test using emulation at the agency's site.

e. The regulation also specifies detailed procedures for an agency to request a waiver from the prescribed policies and procedures.

f. Government agencies that issue a Request for Proposals after the effective date must follow the specified policies and procedures, and may disqualify vendors that do not provide the remote terminal emulation capabilities specified in the solicitation as required by the agency.

2. Objectives and scope.

a. This handbook has two major objectives. The first is to present significant technical information that will help each Government agency (also referred to as a user) to decide if and how to use remote terminal emulation during an ADP system acquisition. The second objective is to define clearly the range of remote terminal emulation capabilities (1) that an agency is permitted to require offerors to provide for benchmark tests during ADP system acquisitions, and (2) that an offeror must have to be qualified to bid on most Federal ADP system acquisitions.

b. To achieve the first objective, this handbook summarizes important benchmarking and remote terminal emulation concepts and terminology, and presents some of the factors and criteria that agencies should consider when deciding whether or not to use remote terminal emulation. It also:

(1) Outlines the types of analyses that should be made during the design and development of emulation benchmark tests;

(2) Defines the data elements and possible formats an agency should use to describe TP workloads for emulation benchmark tests;

(3) Recommends a glossary of relevant terminology;

(4) Includes a bibliography of significant technical and policy materials; and

(5) Cites sources of intragovernment assistance.

c. To achieve its second objective, this handbook gives specifications that define a broad range of remote terminal emulation capabilities. The specifications address the functional aspects (1) of representing the workload demands imposed by remote users and various types of TP devices, and (2) of conducting a benchmark test involving, potentially, multiple RTE's and real terminals. The specifications also cover the physical benchmark test facilities that vendors may need for Federal acquisitions; sample facilities include the number and characteristics of data communication links connecting a SUT to one or more RTE's, and the number and types of real terminals that may be needed concurrently with RTE's. The minimum acceptable accuracy and precision for representing TP workload demands with an RTE are included, as are the definitions and minimum acceptable accuracy and precision of the performance measures produced. In addition, the specifications define (1) minimum RTE log file contents, (2) RTE log summarization and reporting capabilities, and (3) the contents and the physical and logical formats of an RTE log file summary tape that agencies can require vendors to provide.

d. The material in this handbook covers only those aspects of benchmarking that are directly related to the successful use of remote terminal emulation during an ADP system acquisition. Many critical acquisition and benchmarking concepts, policies, procedures, steps, etc. are not discussed in this handbook. Agency personnel, therefore, should study and follow all applicable policy, regulations, procedures, and guidance on benchmarking, including, in particular, "Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment," FIPS PUB 42-1 (subsequently referred to as FIPS PUB 42-1). This handbook supplements FIPS PUB 42-1 in the area of remote terminal emulation. Appendix D provides a bibliography of other relevant policy and technical materials.

3. Structure and audience.

a. Excluding this introductory chapter, this handbook is structured into four chapters and four appendicies. All discussions concentrate on those areas most important to remote terminal emulation. Chapter 2, Benchmarking and Remote Terminal Emulation, introduces these two topics and presents fundamental technical definitions and concepts. Chapter 3, Benchmarking Goals, discusses seven fundamental, but often conflicting goals that a Federal agency should attempt to achieve. The ultimate success of an ADP system acquisition greatly depends on the degree to which these goals are achieved. Chapter 4, Procedural Guidance For Benchmarking, presents specific technical information, suggestions, and recommendations that will assist agencies decide if and how to use remote terminal emulation. This chapter addresses four steps in the benchmarking process that are particularly important to the successful use of remote terminal emulation:

(1) Development of the benchmarking strategy,

(2) Preparation of teleprocessing elements for benchmark tests,

(3) Preparation of Live Test Demonstration (LTD) documentation, and

(4) User-vendor communication.

b. Chapter 5, Remote Terminal Emulation Specifications, defines the emulation capabilities that Government agencies are permitted to require vendors to provide for benchmark tests during ADP system acquisitions. The specifications are divided into six parts:

(1) Teleprocessing device representations,

(2) Terminal operator representations,

(3) Data communication link representations;

(4) RTE driver characteristics;

(5) RTE monitor characteristics;

(6) RTE log analyses.

c. The four appendices contain a reference list of mandatory provisions of this handbook, an example RTE scenario with a sample dialogue and implementation instructions, a recommended technical glossary, and a bibliography.

d. This handbook contains both mandatory procedures and optional suggestions and recommendations concerning how to use remote terminal emulation. Each agency is free to adopt or reject each of the optional suggestions and recommendations. As specified in the temporary regulation, however, each agency must follow the mandatory procedures in this handbook unless that agency obtains a waiver from GSA. Mandatory procedures can be readily identified by the formats of their presentation. The mandatory procedures contain the phrase "It is mandatory that" or "agency shall."

e. The intended audience of this handbook includes both agency and vendor personnel. All readers should study chapter 2 because it presents definitions and concepts that are needed to understand the remainder of this handbook. Agency personnel should study the remaining chapters according to their roles in the acquisition effort. The director of an agency's acquisition program and the manager of the agency group preparing the Request For Proposals needs to study only the temporary regulation and parts of chapter 3. The manager of the agency's benchmarking team and each team member, however, should study all of chapters 3-5, as well as all the appendices. Vendor management and technical personnel should primarily study the temporary regulations, the glossary, and specifications contained in chapter 5. Vendor personnel will benefit from reading the remainder of this document, however, because it will help them understand the benchmarking goals, approaches, terminology, and documentation used by agencies.

4. Intragovernment assistance.

Government agencies can obtain assistance in interpreting and/or complying with this handbook from several sources within the Federal Government. The proper source of intragovernment assistance often depends on the affiliation of the requesting agency. Figure 1-4 lists sources of intragovernment assistance currently available to agencies.

5. Comments and revisions.

a. GSA will review and periodically revise this handbook as needed to reflect changing TP technology, as well as to incorporate additional practical experiences gained through increased Government and industry use of emulation. Revisions to this handbook will be announced by GSA Bulletin, FPR series. GSA will publish and distribute any changes to the mandatory procedures or emulation specifications contained in this handbook at least 120 days before the changes take effect. A notice of the availability of such changes will be published in the Commerce Business Daily. Amendments or revisions to regulator provisions will be promulgated in the Federal Register, under Title 41 CFR Part 1 (the FPR).

b. Interested parties are encouraged to submit comments and suggested improvements to:

General Services Administration
Automated Data and Telecommunications Service (CDD)
Washington, DC 20405
Telephone: (202) 566-1076
FTS 566-1076

AGENCY	SOURCE OF ASSISTANCE
All	General Services Administration Automated Data & Telecommunications Service (CDD) 18th & F Streets, NW, Rm G229 Washington, DC 20405 Telephone: (202) 566-1076 FTS 566-1076
All	Federal Computer Performance Evaluation and Simulation Center Directorate of System Evaluation FEDSIM/NA Washington, DC 20330 Telephone: (202) 274-7910 AUTOVON 284-7910
All	National Bureau of Standards Institute for Computer Sciences and Technology Center for Computer System Engineering (ATTN: Dr M. D. Abrams) Washington, DC 20234 Telephone: (301) 921-3517 FTS 921-3517
Air Force	Air Force Computer Acquisition Center AFCAC/SY Hanscom AFB, MA 01731 Telephone: (617) 861-5265 FTS 844-5265 AUTOVON 478-5265

Figure 1-4. Sources of intragovernment assistance (Part 1 of 2)

AGENCY	SOURCE OF ASSISTANCE
Army	HQDA (ACSA-SD) Pentagon Washington, DC 20310 Telephone: (202) 697-4127 AUTOVON 227-4127
Navy	Department of the Navy ADPE Selection Office (ADPESO) (ATTN: Commander B. Gold) Washington, DC 20376 Telephone: (202) 697-1106 AUTOVON 227-1106

Figure 1-4. Sources of intragovernment assistance (Part 2 of 2)

TABLE OF CONTENTS

CHAPTER 2. BENCHMARKING AND REMOTE TERMINAL EMULATION

TABLE OF CONTENTS CONTINUED

CHAPTER 2. BENCHMARKING AND REMOTE TERMINAL EMULATION

1. Scope. This chapter introduces benchmarking and remote terminal emulation, and presents technical definitions and concepts that are fundamental to understanding the remainder of this document.

PART 1. OVERVIEW OF BENCHMARKING

2. General.

a. Benchmarking is defined in this document as the process of experimentally imposing a test workload on a set of ADP system components to determine selected execution characteristics of the component(s). A test workload, called a benchmark mix, is a collection of user workload elements that typifies the processing environment under evaluation and may range from a single batch program to a combination of many batch programs, test data files, and interactive commands. Typical execution characteristics include computational accuracy, throughput, turnaround time, and response time. A benchmark test is a specific collection of elements (e.g., benchmark mix, execution procedures) used to determine specific execution characteristics of a set of system components, and a benchmark mix execution is a single execution of a specific benchmark mix on a given set of system components. The set of system components evaluated by a benchmark test is called the System Under Test (SUT).

b. Benchmarking is one of several performance evaluation techniques that can help an organization maintain the stability and service quality of ADP systems while managing system change. Organizations typically should use benchmarking in conjunction with other evaluation techniques. Unlike other evaluation techniques, however, benchmarking combines aspects of both performance measurement and performance prediction. Benchmarking can help an organization to:

(1) Determine whether, how well, and at what cost an existing system meets its current ADP needs;

(2) Predict whether, how well, and at what cost an existing system will meet future requirements; and

(3) Predict whether, how well, and at what cost a new or modified system will meet current and/or future ADP needs.

c. The value and accuracy of benchmarking results can surpass other performance evaluation techniques. However, benchmarking costs often exceed the costs of other performance evaluation techniques. Benchmarking, therefore, should be used judiciously and only to satisfy management objectives where the value and necessary accuracy of the results clearly justify the expense.

3. Management objectives.

a. Practical use has demonstrated that benchmarking can be a successful and cost-effective technique for satisfying certain management objectives. The most common of these objectives fall into nine broad categories:[1]

(1) Acquisition evaluation,

(2) Design analysis,

(3) System integration,

(4) Component certification,

(5) Service quality determination,

(6) Stress load analysis,

(7) Regression testing,

(8) Performance improvement, and

(9) Migration planning.

b. Benchmarking is an important part of the ADP acquisition process, both in and out of the Federal Government, and is regularly used to help determine which of several vendor-proposed computer systems should be procured to meet ADP requirements. A benchmark test can provide a quantified, static, and transportable reflection of an organization's ADP requirements. The test, therefore, can be the basis of an equitable comparison of the costs and performance characteristics of various hardware and software alternatives. For many years, benchmarking has been used in the acquisition evaluation of batch processing systems. Today, it is used increasingly with teleprocessing (TP) systems, because the modularity and variety of these systems have increased both the number and complexity of alternatives that must be evaluated and the cost and performance risks that are faced by the user during acquisition.

---

[1] T. F. Wyrick, "Benchmarking Distributed Systems: Objectives and Techniques," in Performance of Computer Installations, ed. D. Ferrari (New York, NY: North-Holland, 1978).

4. Benchmarking framework.

a. Benchmarking, like other testing efforts, should be conducted within an orderly, established framework if it is to be successful. Shetler describes a good general framework for benchmarking (which she refers to as "controlled testing").[2] The benchmarking framework briefly described below has been adapted from Shetler's paper.

b. Figure 2-4 illustrates the general framework within which benchmarking should be conducted. Benchmarking objectives should be understood clearly and defined early, because they are the basis for all decisions that follow. Because benchmarking can be expensive, the expected cost and value of the results should be compared before implementation begins. Based upon the objectives, one or more testable hypotheses are developed. One or more benchmark tests are then designed to prove or disprove the hypotheses. Each design should specify the elements of the benchmark mix, the ADP system components to be tested, the execution characteristics that are to be monitored, and the techniques to be used to impose the test workload(s), to record the relevant execution characteristics, and to analyze the results. The benchmark tests should then be conducted by preparing the benchmark mix, assembling the system components, executing the mix, and recording the results. The results are analyzed, compared to the hypotheses, and documented. More tests are proposed, if necessary, to satisfy the benchmarking objective(s).

5. Functional and capacity tests. Two basic groups of benchmark tests are used for acquisition evaluation, functional tests and capacity tests. A functional test is used to determine if a SUT can accomplish a specific user work item without regard to completion time and other workload demands. For example, a user can employ functional capability tests to determine if a proposed system can read and write a certain tape format, can communicate with a certain make and model interactive terminal, or can remove a portion of main memory without interrupting normal processing. Such a test is sometimes referred to as a functional demonstration or a capability demonstration. A capacity test, in contrast, is used to determine if a SUT can accomplish a specific, often large, set of user work items at a required level of perform-

---

[2] A. C. Shetler, "Controlled Testing for Computer Performance Evaluation," in 1974 National Computer Conference (Montvale, NJ: AFIPS Conference Proceedings, May 1974), pp. 693-699.

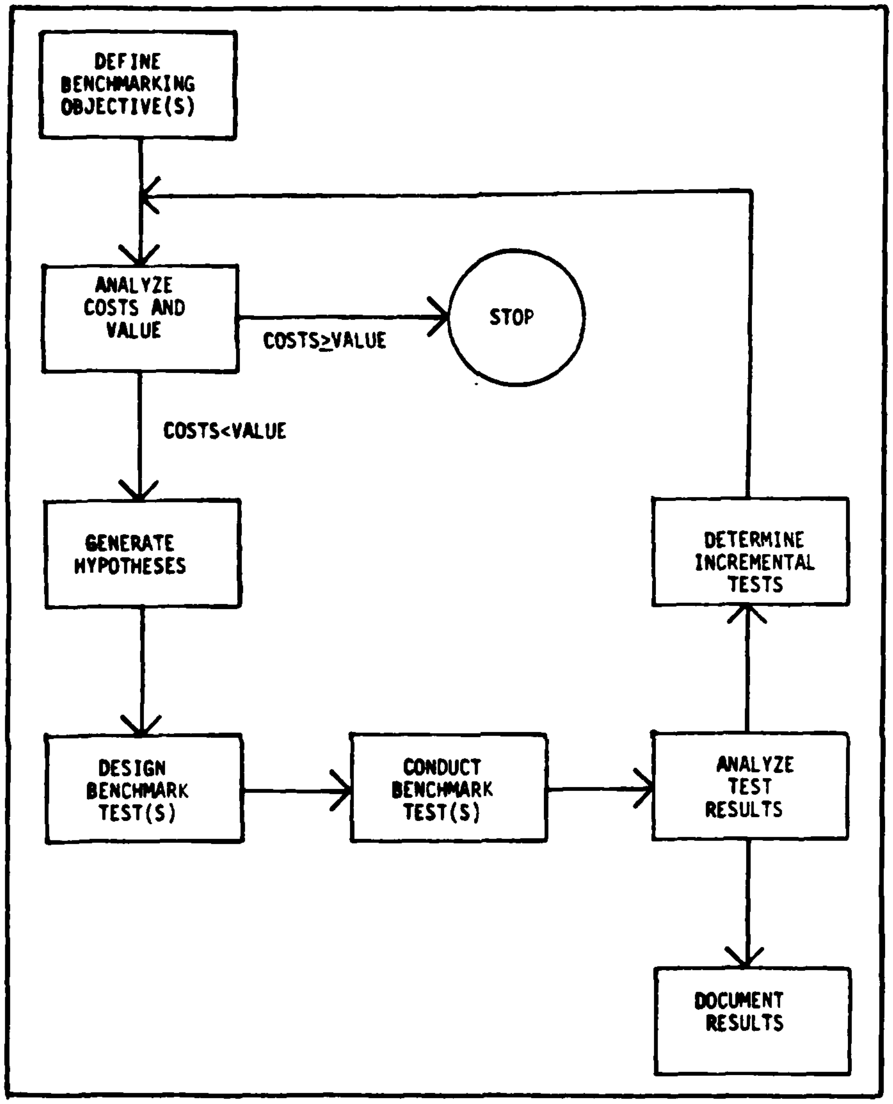


Figure 2-4. Benchmarking framework

ance. A user could employ a capacity test, for example, to determine if a SUT could support a certain number of time-sharing users and maintain acceptable response times. A capacity test is sometimes referred to as a load test or a timed test.

6. Live test demonstration. During acquisition, the user typically requires the vendor to perform certain user-witnessed activities necessary to complete the benchmark tests. Associated with these vendor activities are complimentary user activities; e.g., timing a benchmark mix execution, examining source code for vendor modifications. The period of time during which all these activities occur is called the Live Test Demonstration (LTD).

7. Conclusion. Benchmarking is one of many performance evaluation techniques available to help better manage ADP systems; other techniques include hardware and software monitors, system accounting log analysis, analytic models, and simulation. Each technique has advantages and disadvantages. Benchmarking is not the answer to all ADP management problems, but if used wisely, it can be a cost-effective approach that complements other performance evaluation techniques.

PART 2. OVERVIEW OF REMOTE TERMINAL EMULATION

8. General.

a. Remote terminal emulation today is the principal technique for conducting a benchmark test of a TP system when it would be impractical to conduct the test with the total proposed network of computers, terminal devices, and data communication facilities. Remote terminal emulation uses an external "driver" computer system to impose TP workload demands on the SUT. Potentially, many human operator and remote device characteristics (e.g., interactive, transaction, and batch terminals) and actions can be represented precisely by the driver system in real time. The driver computer system can exchange control and application data transmissions with the SUT through the SUT's operational data communication hardware and software. Remote terminal emulation can use large numbers (up to several hundred) of data communication links of the same speeds, and with the same communication protocols, as an operational environment. When remote terminal emulation is properly used, the SUT cannot distinguish if a real or emulated device is generating the workload.

b. A monitor external to the SUT is a required component of remote terminal emulation. The monitor records on a log file certain aspects of the interaction between the driver and the SUT. Such log files typically include all application data characters transmitted or received by an emulated device and the time each transmission was sent or received by the driver. Data reduction software produces various SUT performance measures (e.g., turnaround time, response time) from the log file after the test. A Remote Terminal Emulator (RTE) is a specific hardware and software implementation of such a driver system. A monitor is usually an integral part of an RTE.

9. Phases. During a competitive TP acquisition, remote terminal emulation consists of five phases: (a) Scenario development, (b) script development, (c) SUT stimulation, (d) scene monitoring, and (e) performance determination. (See figure 2-9.) A benchmark test using remote terminal emulation is often called an emulation benchmark test.

a. Scenario development. A scenario is a vendor- and machine-independent description of some portion of the TP test workload demands. A set of scenarios comprises the entire TP test workload to be represented in a benchmark test. All TP user and device inputs, actions, pauses, and

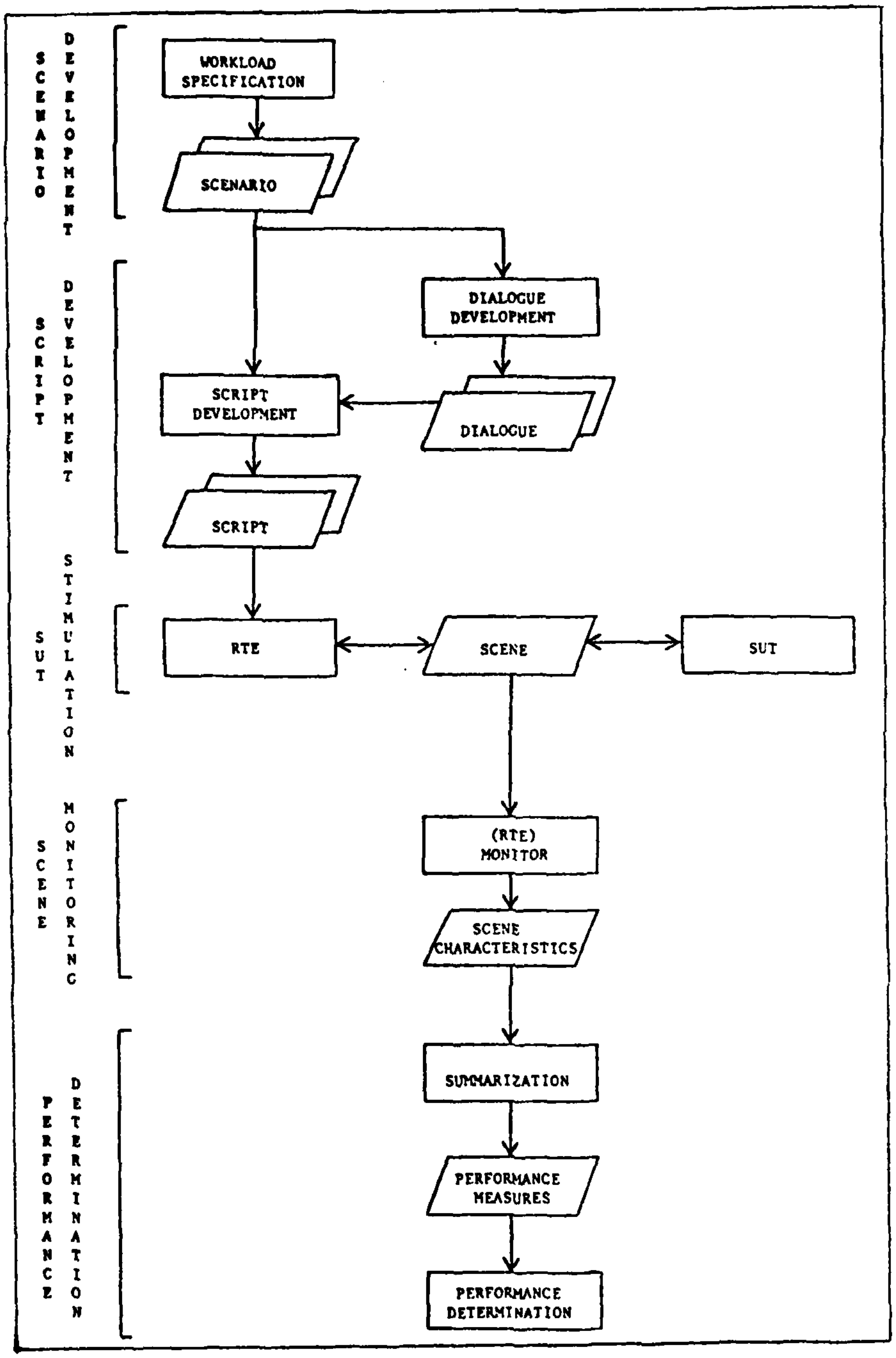


Figure 2-9. Benchmarking using remote terminal emulation

decisions are specified in the scenarios. The scenarios are based on an analysis of current and future TP support requirements; i.e., workload definition. The scenarios are given to all prospective vendors in a common format.

b. Script development. An emulation script is the set of instructions, data, and procedures that causes a particular RTE to impose specific test workload demands on a given SUT. A script depends on the specific RTE, SUT, and scenario used in a benchmark test, because a script includes both (1) the commands to control the RTE and (2) the set of user actions and inputs that will impose on some SUT the TP demands specified in a scenario; i.e., the dialogue. Dialogues often include user LOGON commands, system calls, input transactions, etc. Dialogues, thus, depend on both the SUT and the TP devices represented in a benchmark test. (A dialogue is used with a "real" terminal, as well as with an RTE, when a benchmark test includes the use of both emulated and real terminals.) A different script and dialogue are usually produced from each scenario for each vendor-proposed SUT. A set of scripts comprises the test workload imposed by the RTE during a benchmark test. Vendors usually produce both dialogues and scripts from user-supplied scenarios.

c. SUT stimulation. SUT stimulation is the use of the RTE to impose test TP workload demands on the SUT. The RTE is controlled by a set of scripts. The dynamic interaction between the RTE and the SUT during stimulation is called the scene.

d. Scene monitoring. Scene monitoring is the recording of certain characteristics of the scene. The scene characteristics needed for performance determination are chosen for logging. Possible scene characteristics are all application data characters transmitted or received by an emulated device and the time each transmission was sent or received by the RTE. Scene monitoring may be provided by the RTE and/or by other recording devices independent of the SUT.

e. Performance determination. Performance determination involves (1) summarizing recorded scene characteristics and (2) using these summaries to evaluate the performance of the SUT with respect to the test workload. Summary information is provided by scene data reduction programs. The same performance measures are used for all vendors and SUT's. Preliminary performance determination usually is made at the conclusion of the benchmark test, and an adequate audit trail is provided to allow for a subsequent final determination.

10. Test workload components.

a. To use remote terminal emulation for acquisition benchmark tests, procuring agencies must describe and interrelate to prospective vendors at least four basic TP workload components:

(1) Terminal operator actions; e.g., input, output, decisions, rates;

(2) SUT-network interface characteristics; e.g., TP devices, links, protocols;

(3) The software with which the terminal operators interact; e.g., vendor-proposed text editors, Government-supplied applications; and

(4) The data files accessed and/or created by the operators.

b. The general relationship of these four components is shown in figure 2-10.

11. Performance measures. Three basic performance measures (throughput, turnaround time, and response time) are provided by all RTE's that conform to the functional specifications in this document. These measures, as defined herein, are well-defined and technically consistent for all vendor systems. Detailed definitions of these measures are contained in chapter 5, part 6.

a. Throughput. As used in this handbook, throughput is defined as the number of user work items successfully completed within a predefined time interval. To compute throughput, agencies and vendors must be able to (1) count the number of work items successfully completed and (2) record both the time that work began on the first item and the time that work ended on the last item. A clearly defined start and end of a benchmark mix execution are needed to calculate throughput.

b. Turnaround time. Turnaround time is used in this handbook to refer to the time interval between the initiation of a user work item and the successful completion of the work item. To calculate turnaround time, agencies and vendors must be able to observe clearly the start and end of work on each item and must be able to record the times of the start and the end.

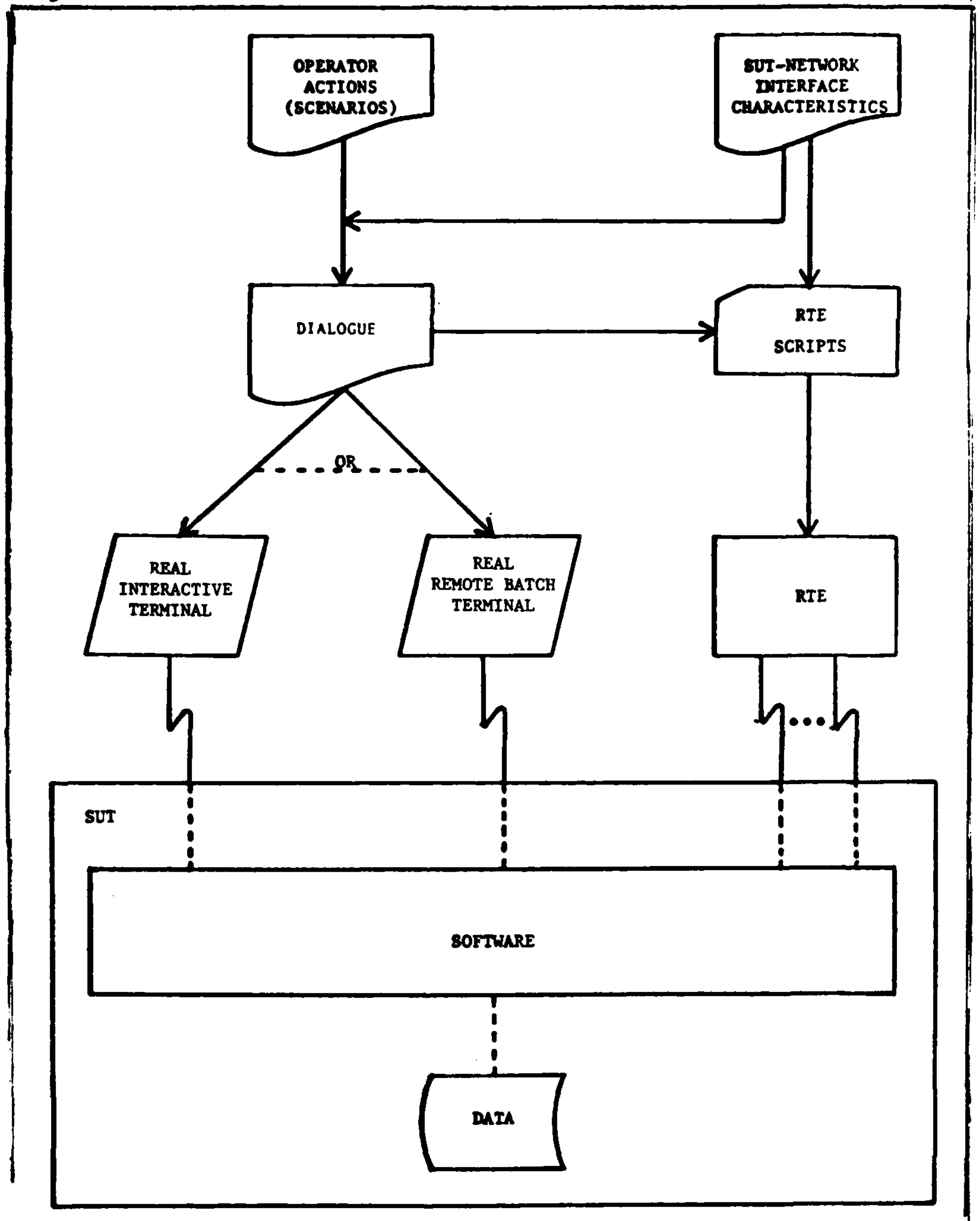


Figure 2-10. Test workload components for benchmark tests using remote terminal emulation

c. Response time. Response time, defined only for interactive work units, refers to the elapsed time from the last keystroke of an operator input at an interactive device until the first printable character of the resulting SUT response appears at the user's device. Response time is the most difficult performance measure to define and use precisely and consistently for all vendors, partly because of the enormous variability of interactive devices, application types, protocols, etc.

12. Data communication input-output pair.

a. General.

(1) An important concept that one must understand in order to use this document is the concept of the data communication input-output pair, or I/O pair. Conceptually, an I/O pair is an exchange of functionally related data transmissions by an emulated device and the SUT. Either a SUT or an emulated device can initiate an I/O pair. Some I/O pairs perform only overhead control functions; e.g., a poll sent by the SUT and a negative acknowledgement returned by the emulated device. Many I/O pairs, however, are explicitly related to accomplishing user functions; e.g., a single timesharing command and the resulting SUT output. During benchmark tests, these user-function I/O pairs (referred to as application I/O pair(s)) are of primary concern.

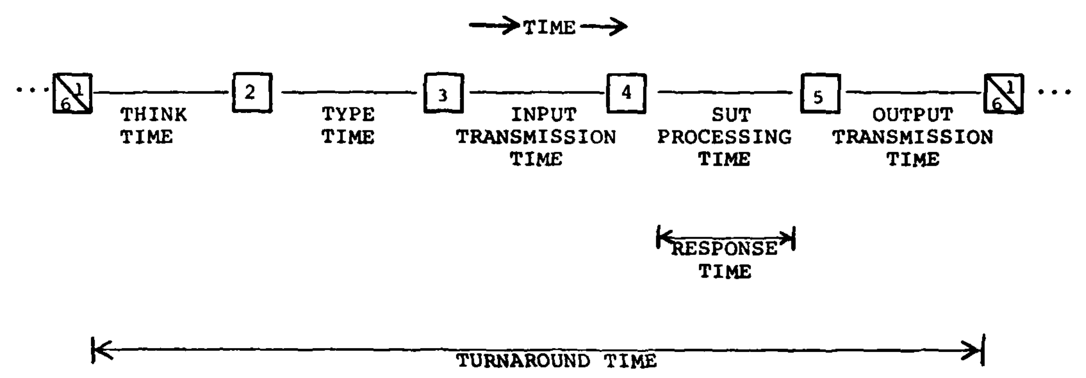
(2) Teleprocessing scenarios are system-independent descriptions of user TP workload demands to be performed during a benchmark test, and are expressed as some number of user functions. To execute a benchmark test, these user functions must be translated into specific user actions (e.g., keystrokes, submissions of remote batch card decks) and, ultimately, into application I/O pairs; e.g., command and response. The nature and number of application I/O pairs needed to perform a given user function vary from vendor to vendor and from system to system. The general definitions of application I/O pairs, however, are fundamental to the performance measures used to evaluate benchmark test results and underlie the RTE specifications described in this document. These general definitions are outlined below. Chapter 5, part 5 contains more precise statements of the I/O pair events that vendor RTE's must time-stamp. (The ultimate technical definitions of I/O pairs, however, depend on unique vendor hardware and software.)

b. Asynchronous interactive devices. Figure 2-12.1 illustrates the general definition of an application I/O pair for asynchronous interactive devices. The relationships of this definition to turnaround time, response time, think time, and type time are shown. Only one line of input data is transmitted to the SUT in each application I/O pair. The total SUT output resulting from a single user input may be only a non-printable control character (e.g., carriage return) or may be several lines of data. The I/O pair ends when the emulated device receives the last character of the resulting SUT output.

c. Synchronous interactive devices. The general definition of an application I/O pair for synchronous interactive devices, and the relationships of response time and turnaround time, are illustrated in figure 2-12.2. A single user input can result in more than one transmission block being sent to the SUT, and/or the resulting SUT output may contain several transmission blocks. Print time also may overlap output transmission time when the output contains several blocks. Print time may or may not be defined for certain synchronous devices and/or user functions.

d. Remote batch terminals. Figures 2-12.3 and 2-12.4 illustrate the I/O pair definitions for remote batch terminals performing card input and print output, respectively, and show the typical simultaneity of operations. The relationship of turnaround time is shown, but response time is not defined for this TP device type.

e. Remote host systems. Figures 2-12.5 and 2-12.6 illustrate the I/O pair definitions for remote host systems performing file and batch job input and file and batch job output, respectively. Again, turnaround time is shown, but response time is not defined for this TP device type.

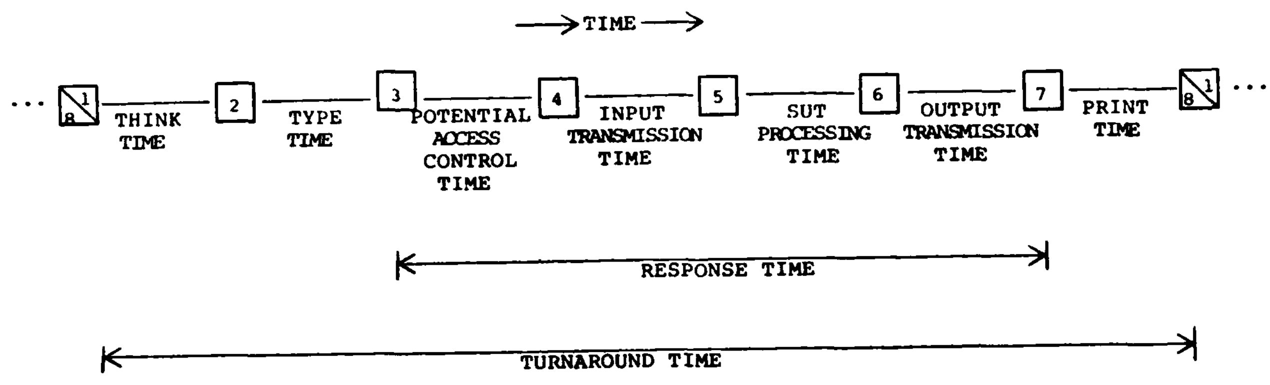


EVENTS:

1. Start of any specified user think time; Start of I/O pair
2. End of think time; Start of any specified type time
3. Transmission by device of the first character of the user input
4. Transmission by device of the last character of the user input
5. Receipt by the device of the first printable character of the SUT output
6. Receipt by the device of the last character of the SUT output; End of I/O pair

NOTE: More precise definitions of the events that comprise this input-output pair are contained in chapter 5, part 5.

Figure 2-12.1. Application input-output pair for asynchronous interactive devices

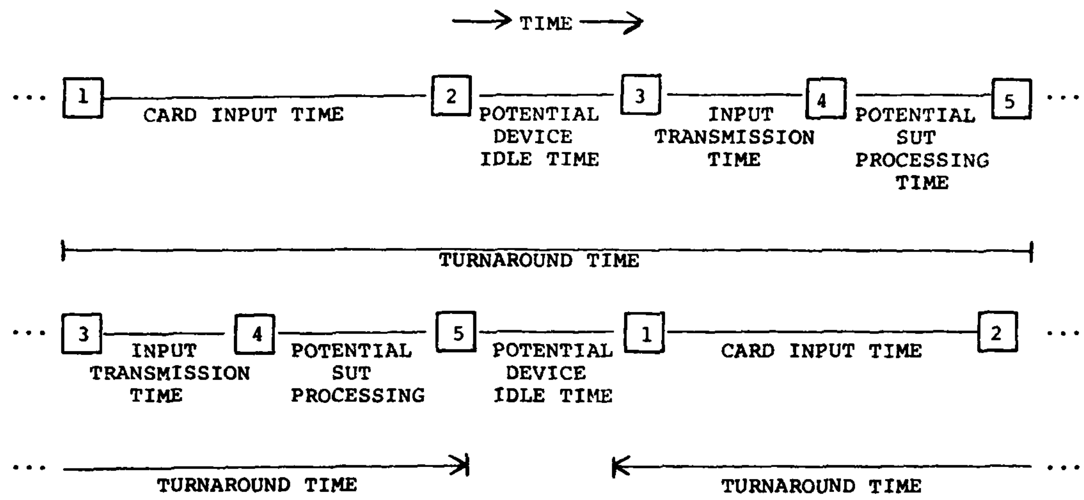


EVENTS:

1. Start of any specified user think time; Start of I/O pair
2. End of think time; Start of any specified type time
3. Last keystroke of user input; e.g., transmit key, enter key
4. Transmission by device of the first character of user input
5. Transmission by device of the last character of user input
6. Receipt by the device of the first character of the first error-free transmission block of the SUT output
7. Receipt by the device of the last character of the first error-free transmission block of the SUT output; Start of any specified print time
8. End of any specified print time, or device ready to accept next user keystroke (e.g., keyboard unlock), which ever occurs last; End of I/O pair

NOTE: More precise definitions of the events that comprise this input-output pair are contained in chapter 5, part 5.

Figure 2-12.2. Application input-output pair for synchronous interactive devices

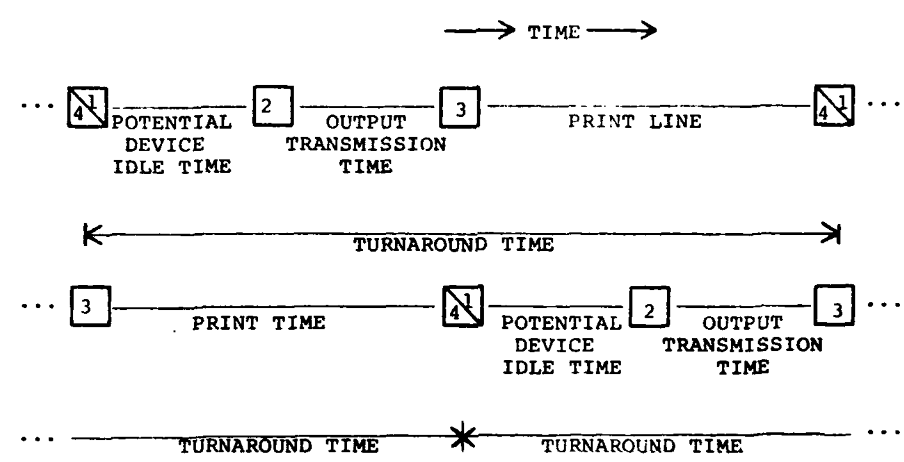


EVENTS:

1. Start of any specified card input time; Start of I/O pair
2. End of card input delay; Input transmission block ready to be sent to SUT
3. Transmission by the device of the first character of input transmission block
4. Transmission by the device of the last character of input transmission block
5. Receipt by the device of a SUT output acknowledging correct receipt of input transmission block, <u>if</u> SUT sends acknowledgement of every block; End of I/O pair

NOTE: More precise definitions of the events that comprise this input-output pair are contained in chapter 5, part 5.

Figure 2-12.3. Application input-output pair for remote batch terminals performing card input

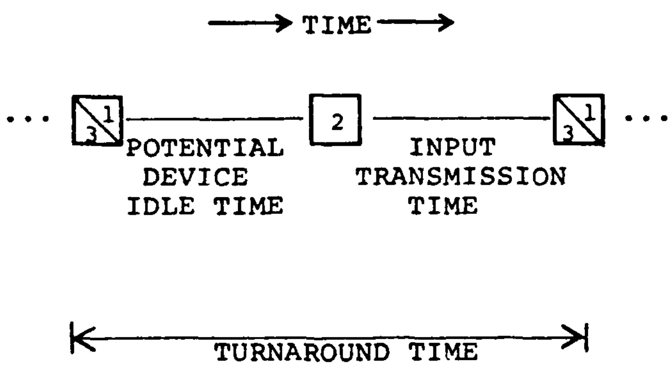


<u>EVENTS</u>:

1. Receipt by the device of the first character of SUT transmission requesting to initiate print output, <u>if</u> this is start of an output function, or expiration of previous print time; Start of I/O pair
2. Receipt by the device of the first character of an error-free output transmission block
3. Receipt by the device of the last character of an error-free output transmission block; Start of print time
4. Expiration of print time; End of I/O pair.

NOTE: More precise definitions of the events that comprise this input-output pair are contained in chapter 5, part 5.

Figure 2-12.4. Application input-output pair for remote batch terminals performing print output

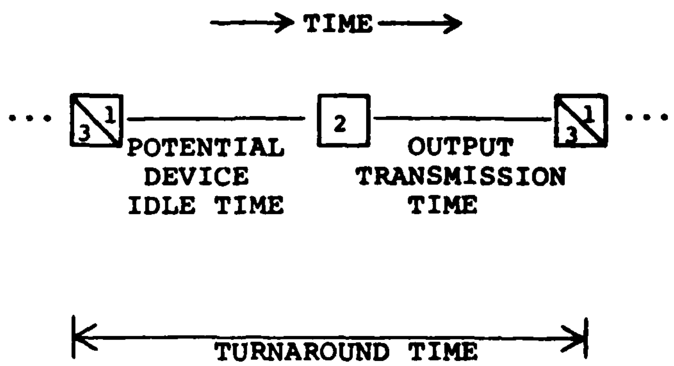


<u>EVENTS</u>:

1. Transmission by the device of either (a) the first character of a message requesting to initiate a file input, if this is the start of an input operation, or (b) the last character of the previous file input transmission block; Start of I/O pair
2. Receipt by the device of a transmission from the SUT acknowledging the correct receipt of the previous file input transmission block, if the SUT acknowledges every block; Transmission by the device of the first character of the input transmission block
3. Transmission by the device of the last character of the input transmission block; End of I/O pair

NOTE: More precise definitions of the events that comprise this input-output pair are contained in chapter 5, part 5.

Figure 2-12.5. Application input-output pair for remote host systems performing file or batch job input



EVENTS:

1. Receipt by the device of either (a) the first character of a message from the SUT requesting to initiate a file output, if this is the start of an output operation, or (b) the last character of the previous error-free output transmission block; Start of I/O pair
2. Receipt by the device of the first character of an error-free output transmission block
3. Receipt by the device of the last character of an error-free output transmission block; Transmission by the device of a message acknowledging the correct receipt of the block, if an acknowledgement is sent for every block; End of I/O pair

NOTE: More precise definitions of the events that comprise this input-output pair are contained in chapter 5, part 5.

Figure 2-12.6. Application input-putput pair for remote host systems performing file or batch job output

TABLE OF CONTENTS

CHAPTER 3. BENCHMARKING GOALS

Figure 3-1. Benchmarking goals

CHAPTER 3. BENCHMARKING GOALS

1. Scope and audience.

a. When benchmarking is used during a competitive ADP system acquisition, the procuring agency (also referred to throughout this handbook as the user) should attempt to achieve each of the seven fundamental benchmarking goals listed in figure 3-1. All of these goals, however, cannot be totally achieved, because they usually conflict. For example, a highly representative benchmark test can be extremely costly and time consuming for both the user and competing vendors; the required use of certain benchmark test options, sometimes necessary to increase the accuracy and precision of system sizing, can effectively exclude from competition those vendors that do not have the needed benchmarking facilities or expertise.

MINIMIZE TIME AND COST OF ACQUISITION
MAXIMIZE COMPETITION
MAXIMIZE QUALITY OF SYSTEM SIZING
MAXIMIZE BENCHMARK REPRESENTATIVENESS
MINIMIZE BENCHMARK DISCREPANCIES
MAXIMIZE BENCHMARK UNIFORMITY
MAXIMIZE BENCHMARK REPEATABILITY

Figure 3-1. Benchmarking goals

b. The ultimate success of an ADP system acquisition greatly depends on the degree to which benchmark tests achieve these seven goals. For example, highly inaccurate system sizing may result in the acquisition of a system that fails to meet the user's needs, and benchmark tests that produce different workload demands for each vendor (i.e., low uniformity) may result in vendor protests, delay, and/or cancellation of the acquisition. Moreover, most strategic and operational decisions about the necessity, design, implementation, and conduct of benchmark tests affect the degree to which these goals are achieved. It is critical, therefore, that agency personnel participating in an acquisition understand these goals and the need to compromise and reconcile the inevitable conflicts. Agency management, in particular, must understand these conflicting goals, because a critical management function is to establish the combination of goal levels that best insures the overall success of the acquisition.

c. This chapter describes each of these goals in the order listed in figure 3-1. Later chapters compare benchmarking and remote terminal emulation alternatives by outlining the relative effects of the alternatives on the achievement of these goals. One should read the goal descriptions according to one's role in the acquisition effort. The director of an agency's acquisition program and the manager of the Request For Proposals (RFP) development group should concentrate on the first four goals. The manager of the agency's benchmarking team and each team member, however, should study all seven goal descriptions. Familiarity with these goals and their terminology will help vendor personnel communicate with procuring agencies and understand the remainder of this document.

2. <u>Minimize time and cost of acquisition</u>.

a. The competitive acquisition of an ADP system can require very large investments of time and money by the user and each competing vendor. In addition to the price of the procured system, a user can incur costs for such user activities as:

(1) Workload definition;

(2) Analysis of technical and procurement alternatives;

(3) Benchmarking;

(4) Administrative review and approval;

(5) Preparation and release of procurement documents; e.g., Request For Proposals;

(6) Interaction with vendors;

(7) Technical and cost evaluation of vendors' proposals;

(8) Negotiations and award;

(9) Training;

(10) Site preparation;

(11) Installation;

(12) Conversion; and

(13) Contract administration.

b. A fundamental user goal should be to minimize the time and cost of the acquisition. If the acquisition takes too much time, a user may be unable to satisfy ADP mission requirements because the new system would not be available when needed. In addition, additional time often leads to increased user costs. Minimizing the total cost for the procured system and all acquisition activities is a basic tenant of good management and is critical to the efficiency and effectiveness of the procuring agency.

c. In addition to the costs of developing and/or manufacturing the system hardware and software, a vendor can spend time and money on such vendor activities as:

(1) Analysis of user-provided procurement documents;

(2) Determination of the system configuration(s) to bid;

(3) Interaction with the user;

(4) Preparation and submission of a proposal;

(5) Benchmarking;

(6) Negotiation;

(7) Installation;

(8) Conversion;

(9) Acceptance testing; and

(10) Contract administration.

d. Vendor costs are eventually passed on to users as higher prices. During a competitive acquisition, a vendor usually attempts to minimize the time and funds expended in order to reduce his bid price. A vendor often declines to bid when the acquisition time and cost incurred before contract award (i.e., the "entry fee") are determined to be too great. A user who conducts an acquisition that requires large "entry fees" from participating vendors, therefore, increases his direct and indirect acquisition costs and also risks a reduction in the probable number of vendors who will submit bids.

e. Benchmarking can require significant investments of time and money by the user and each competing vendor. Benchmark tests using remote terminal emulation, moreover, usually involve more time and higher costs than tests using only batch programs and/or a very small number of real terminals. For the user, the greater time and costs are primarily due to the increased complexity of the test design, the need to develop and test scenarios, and the more detailed test documentation required. For a vendor, the greater time and costs are principally due to the additional hardware, software, and machine time needed, the number of RTE hardware and software changes required for each acquisition, and the need to develop and test RTE scripts.

f. The usage guidance and RTE specifications contained in this handbook will help reduce the time and cost of emulation benchmark tests, because the guidance and specifications will (1) improve communications between the user and vendors, (2) lead to better test designs and documentation, and (3) substantially reduce RTE hardware and software changes that vendors will need to make for future Federal acquisitions.

g. The benchmarking time and costs that are appropriate for an acquisition depend upon the specific circumstances of that acquisition. The user should carefully balance the expected time and costs of benchmarking with the importance placed on other benchmarking goals.

3. <u>Maximize competition</u>.

a. Federal ADP acquisition policy specifies that agencies should strive for the maximum practical competition. Competition reduces the prices bid by competing vendors and encourages innovation and continued growth in the ADP industry. Most user requirements can, if phrased properly, be totally satisfied without reducing competition. Full competition is not always possible, however, because there are valid user needs that some vendors are either unable or unwilling to meet. The degree of competition that is appropriate and practical for an acquisition depends upon the circumstances of that acquisition and should be carefully evaluated by the user.

b. Benchmark tests can reduce competition, primarily because of the time and cost to vendors. Benchmarking, however, may be necessary to satisfy other user acquisition goals. When a benchmark test is employed during ADP acquisition, the user should (1) insure that the importance of each individual aspect of the test is greater than the time and cost to the user and vendors, (2) design each test requirement to encourage competition, and (3) allow vendors the maximum possible flexibility in complying with test requirements.

c. The procedures on the use of remote terminal emulation expressed in this handbook occasionally may reduce competition by permitting a Federal agency to disqualify any vendor that does not provide the benchmarking capabilities specified in the handbook and required by the agency. The procedures, however, should increase competition over time because they clearly define and limit, except under extraordinary circumstances, the emulation capabilities that agencies could require vendors to provide.

4. <u>Maximize quality of system sizing</u>.

a. During a competitive acquisition, each vendor determines the system configuration(s) to bid by a process called system sizing. System sizing is defined in this handbook as the process of determining a configuration of hardware and software components that can accomplish a specific set of workload demands at a required level of performance. The quality of system sizing depends on both (1) the probability that the resulting configuration can accomplish the target workload demands at the required performance level; i.e., the probability that the sizing is accurate; and (2) the degree to which the capacity of the resulting configuration approaches the minimum capacity needed to accomplish the demands at the required performance level; i.e., the precision of the sizing.

b. The available sizing techniques, in the typical order of increasing sizing quality and cost, are: (1)

Professional judgment based on experience, (2) static modeling, (3) analytic modeling, (4) simulation, and (5) benchmarking. Vendors often use a combination of these techniques during a single acquisition.

c. The quality of each vendor's system sizing is critical to the user. The sizing must be accurate if the user is to reduce the likelihood that the system eventually acquired will fail to satisfy the user's mission requirements at any point during the contractual life of the acquisition; i.e., reduce the user's mission risk. The probability that the sizing is accurate is fundamental to the success of the acquisition. The precision of the sizing affects the likelihood that the user will pay more than is necessary to satisfy its ADP requirements; i.e., the user's cost risk. Greater sizing precision results in less excess capacity in the proposed system configuration, which, in turn, usually reduces the proposed price of the system. While user mission and cost risks can be reduced somewhat by contractual terms and conditions (e.g., excess quantities, value engineering), the lowest risk levels are achieved when the quality of each vendor's system sizing is maximized.

d. The user describes to vendors one or more sets of workload demands that the system to be acquired must be able to accomplish, as well as the acceptable levels of performance for completing the demands. (Vendors use these requirements for system sizing.) In addition to specifying these requirements, a user often intentionally limits the configurations that a vendor can bid by limiting the number, types, characteristics, and/or installation schedules of hardware and/or software components. These configuration constraints typically reflect such factors as industry and Government standards, compatibility with existing components, availability requirements, and operational limitations; e.g., floor space, size of operations staff. For example, a user often specifies the magnitude of change that defines a major system augmentation (e.g., any change in CPU or main memory) and then limits the number of times each vendor can augment the initially proposed configuration.

e. There are several methods by which a user can increase the quality of the system sizings performed by vendors. One method is to impose as few configuration constraints as possible, thereby increasing vendors' flexibility to bid cost-effective systems. The most important method, however, is to describe clearly, unambiguously, and in maximum practical detail the user workload demands and required performance levels. A clear, accurate description of requirements is important because vendors must understand a user's needs for both functional capability and system capacity, throughout the contractual life of an acquisition, before they can size and bid cost-effective systems.

f. A user can describe requirements to vendors by a narrative in an RFP, by benchmark test(s), or by a combination of narrative and benchmark test(s). When generally accepted, vendor-independent terminology is available and is employed carefully, a user can describe requirements in RFP narrative that is clear, concise, unambiguous, and of sufficient detail for high quality system sizing. Such terminology is available for describing most functional capabilities and a few, limited dimensions of capacity; e.g., characters of on-line disk storage, print volume, number and speeds of terminal ports. Many important requirements, however, cannot be described in narrative that vendors can use for high quality system sizing, principally because there is no widely accepted, vendor-independent terminology for unambiguously describing such user work items as "transaction," "command," "job," etc. Such requirements and their interrelationships can best be described (and measured) by benchmarking. In addition, mandatory benchmark tests insure that all vendors employ the same sizing technique, as well as the technique that produces the highest quality sizings. The quality of vendors' system sizings, therefore, is greater when a combination of narrative and benchmarking is used to describe user requirements than when narrative is used alone.

g. Remote terminal emulation is the benchmarking technique that can produce the highest quality system sizing for TP systems. Through emulation, a user can define and impose the workload demands necessary to determine the types and characteristics of the TP-related system components needed to satisfy requirements, including front-end processor speed and memory, CPU resources needed to handle TP overhead processing, etc. Moreover, the use of the features defined in this handbook increases the quality of vendors' sizings done with emulation benchmark tests.

h. The fundamental motivation for using remote terminal emulation during an acquisition is to increase the quality of system sizing. Emulation (indeed all benchmarking), however, increases the preaward time and cost for users and vendors and may decrease competition. In theory, the time and cost of benchmarking should not exceed the cost avoidance obtained by the increased quality of the sizing, unless the time and cost is justified by the critical nature of user mission requirements. This theory is virtually impossible to apply quantitatively, however, because of the difficulty of calculating the savings derived from higher quality sizing, and the direct and indirect costs of benchmarking. In practice, a user subjectively chooses a level of sizing quality that is sufficient to reduce to acceptable limits his perceived mission and dollar risks. A major criterion for deciding if and how to use remote terminal emulation, therefore, is the subjective value to the user of higher quality system sizing, especially of the TP-related system components. For each acquisition, the user should analyze the value of higher quality system sizing and then balance that value with the benefits associated with other user acquisition and benchmarking goals.

5. Maximize benchmark representativeness.

a. Benchmark representativeness is defined as the degree to which a benchmark test duplicates an operational processing environment anticipated to occur during the contractual life of an acquisition. It is essentially impossible to duplicate precisely any operational environment for benchmarking purposes because of the enormous number, complexity, uncertainties, and interdependencies of the human, hardware, software, and workload characteristics involved. User mission and cost risks are minimized, however, when the representativeness of each benchmark test is maximized.

b. The representativeness of a benchmark test depends primarily on three technical factors: (1) The test workload demands and the associated performance levels, (2) the SUT configuration for the test, and (3) the benchmarking technique(s) used. The test workload demands and associated performance levels are important because benchmark representativeness cannot exceed the degree to which all characteristics of the test workload demands (e.g., volumes, types, arrival rates, distributions, and sequences), as well as the desired performance levels for completing these demands, reflect the user's anticipated operational environments. In addition, representativeness decreases when there are differences between the SUT configuration and the configurations proposed and/or ultimately installed at the user site. User risks are reduced when the SUT components and the workload demands that significantly affect SUT cost and performance are included in the benchmark test. Certain benchmarking techniques typically can be used to represent closely some operational processing environments but not other environments. The user should employ the benchmarking techniques that can impose on the SUT the workload characteristics the user has decided to include in the test. One of the most important attributes of the remote terminal emulation technique is that it can be used to represent precisely certain characteristics of TP workload demands that cannot be represented by any other technique.

c. It is often very costly and time consuming for a user and vendors to achieve high representativeness. Representativeness can increase only when a user invests the time and money necessary to increase the detail and thoroughness of the test workload and the complexity of the benchmark test structure, including the benchmarking techniques. To increase representativeness, a vendor must spend the time and money to provide and maintain a more complete SUT configuration, to provide the benchmarking technique(s) required, and to implement the complex test structure.

d. The benchmark representativeness that is appropriate and achievable for an ADP acquisition depends on the specific circumstances of that acquisition and on the purpose of the test. The probable amount of error in the user's workload definition affects the detail and thoroughness of the test workload and, therefore, the maximum achievable representativeness. Some test objectives can be satisfied with less representativeness than others; e.g., functional capability often can be demonstrated with a single, real terminal instead of with a large number of emulated devices. The user should carefully examine and choose the desired representativeness of each benchmark test to achieve the best total combination of benchmark goals; e.g., minimize time and cost, maximize the quality of system sizing.

6. Minimize benchmark discrepancies.

a. To confidently use a benchmark test for vendor comparison and selection, a user must minimize all discrepancies between the manner in which the user intended for the test to be conducted and the manner in which each vendor actually conducted the test. These benchmark discrepancies can be either technical or procedural. A user minimizes these discrepancies by a process called benchmark verification, defined as the act of determining the degree to which a vendor conducted a benchmark test in the manner intended by the procuring agency. Discrepancies can occur for any of the following reasons:

(1) Omissions, ambiguities, inconsistencies, or errors in user-developed benchmark materials;

(2) Misunderstandings by a vendor of the user's intentions and desires;

(3) Unintentional vendor mistakes;

(4) Intentional actions or misrepresentations by a vendor; and

(5) Hardware or software errors.

b. It is practically impossible, however, for a user or a vendor to ascertain with total confidence that there were no discrepancies in the conduct of a test, because of the complexities of modern TP systems and benchmark test designs.

c. To successfully verify a benchmark test, a user must develop a verification strategy. The strategy should reflect the user's answers to several important questions:

(1) Which specific technical and procedural aspects of the test will be examined for verification? Users often examine such technical aspects as batch program source code, data base contents, SUT hardware components and interconnections, and the completion status of each program executed during the test.

(2) For each aspect of the test, what magnitude of discrepancy will be allowed before the test execution is declared invalid? Users, for example, regularly permit vendors to use less SUT hardware for a test than proposed and, sometimes, allow minor variations in component models and options; e.g., fewer tape and disk drives.

(3) What verification techniques will be used and how thoroughly will each be applied? One of the most important techniques for reducing discrepancies is communication between a user and vendor. Many discrepancies, arising both from problems in user-developed benchmark material and vendor misunderstandings, can best be resolved by the user's previewing the benchmark test(s) individually with each vendor in a meeting held after proposals are received but 2 or 3 weeks before the LTD's. Another verification technique is the physical examination of SUT hardware components and

each line of source code. The user must choose the detail of these examinations; e.g., examine only selected source programs, do not trace the cabling between SUT components. The use of remote terminal emulation increases the potential for discrepancies in a benchmark test, primarily because it increases the test complexity. Chapter 4, part 1 outlines verification techniques that can be used during an emulation benchmark test, and chapter 5 includes specifications for RTE functional capabilities intended for use in verification.

d. During the development of a verification strategy, the user should consider the following:

(1) The likelihood of a discrepancy in each aspect of the test, based upon such factors as the complexity of each aspect;

(2) The level of user and vendor effort required to examine each aspect and to employ each verification method;

(3) The level of negative effect of certain discrepancies and verification methods on the attainment of other benchmarking goals; e.g., reduced representativeness, lower quality sizing, increased user and vendor time and cost; and

(4) The subjective level of confidence required by the user to believe that the possible magnitude of discrepancies does not invalidate the test.

e. The verification strategy is critical to the success of a benchmark test and should carefully be developed before finalizing the benchmark test design.

7. <u>Maximize benchmark uniformity</u>.

a. Benchmark uniformity, an important goal in the competitive acquisition process, is the degree of similarity between test workload demands imposed on different SUT's by the execution of the same benchmark mix. So that all vendors are treated equally, a benchmark mix used for acquisition should theoretically impose on all SUT's test workload demands that are identical in all characteristics; e.g., volume, types, arrival rates and sequence. Absolute uniformity is virtually impossible to achieve, however, because of the complexity and diversity of TP systems and the dependence of certain workload characteristics on the system processing the workload. In addition, absolute uniformity conflicts with certain other benchmarking goals held by both the user and the vendors. A user, therefore, should carefully analyze and choose the appropriate uniformity for each test so that all vendors are treated equitably and the best combination of benchmarking goal levels are achieved.

b. Extremely high benchmark uniformity can reduce the representativeness of a test. Certain characteristics of a user's workload in the anticipated operational environment often depend on characteristics of the system ultimately acquired e.g., file structures, physical I/O block sizes, interactive subsystems, interactive commands. Vendors are usually granted permission to make limited modifications to benchmark mixes to reflect such dependencies and to demonstrate possible processing efficiencies; such modifications increase the test representativeness but decrease uniformity. Moreover, it is often difficult for a user that is unfamiliar with a particular vendor's product line to evaluate the effect of such modifications on the quality of system sizing, as well as on the magnitude of benchmark discrepancies.

c. Some benchmarking techniques are inherently less capable of high uniformity than others, and not all implementations of some techniques can represent certain workload characteristics. While remote terminal emulation has the potential for exceptional uniformity, the uniformity actually achieved in the past has been less than desired. This reduced uniformity was caused primarily by the lack of functional similarity among both the RTE's and the physical test facilities provided by different vendors. Moreover, vendors and users today occasionally expend considerable time and funds analyzing, documenting, negotiating, and modifying benchmark test procedures and/or RTE's and test facilities because of a lack of functional similarity. These efforts usually further reduce uniformity. When implemented by vendors and employed properly by agencies, however, the remote terminal emulation capabilities and usage guidance in this handbook will increase greatly the uniformity achieved during benchmark tests.

d. A common benchmark verification approach is for the user to intentionally modify certain workload characteristics of a benchmark mix (e.g., file contents, sequences of interactive dialogues) immediately before the start of the benchmark mix execution or the LTD. The user states the types and magnitudes of such changes when the benchmark mix is initially released. Benchmark uniformity is maintained by insuring that the workload demands of both the benchmark mix initially released and the mix actually used for the LTD are not significantly different.

8. <u>Maximize benchmark repeatability</u>.

a. Benchmark repeatability is the degree of similarity between two executions of the same benchmark mix on the same SUT. Two important factors often used to evaluate repeatability are: (1) The changes in the values of the performance measures employed in the test, and (2) the changes in the test workload demands produced by the benchmark mix. Repeatability should not be confused with benchmark uniformity. Uniformity indicates the change of test workload demands when the same benchmark mix is executed on different SUT's.

b. Neither a user nor a vendor can totally eliminate random differences in the operation of a complex TP system during a benchmark test, because no one can control such hardware, software, and human factors as transient hardware errors, changes in operator typing rates, minute variations in disk arm movement and rotational times, automatic adjustments in paging and scheduling algorithms, etc. These inherent, random differences in SUT operation will always cause the values of benchmark performance measures to vary from mix execution to mix execution, even if all other test factors are identical. Total repeatability, therefore, cannot be obtained. The principal way to maximize benchmark repeatability is for the user and vendor to minimize, from execution to execution, all changes in all the workload characteristics produced by the benchmark mix. A user, however, should not design a benchmark test to intentionally mask any extreme variability in SUT performance that is due to inherent, random differences in SUT operation, because such extreme variability may be indicative of technical deficiencies in the SUT.

c. Low repeatability can significantly decrease the quality of system sizing. To minimize the risk of failing the LTD, a vendor usually configures a system with sufficient capacity to accomplish the test workload demands, at the required performance levels, throughout the range of probable variations. The amount of "extra" capacity needed for "insurance" is indirectly proportional to the amount of repeatability of the benchmark test. Lower repeatability also increases the time and cost to the vendor to prepare for the LTD, which in turn decreases the likelihood that the vendor will participate in the acquisition.

d. A user occasionally can increase the benchmark representativeness and decrease benchmark discrepancies by introducing carefully chosen changes to the test workload characteristics. Because a user's operational environment contains random variability, a benchmark test may be more representative if the test workload included some limited, natural variability; e.g., random think times, random sequencing of transaction inputs. By minor changes to certain workload characteristics, a user also can reduce the likelihood that a vendor has taken improper advantage of those workload characteristics to improve SUT performance. These changes, however, reduce repeatability. Because of the potential negative effect of such changes on the quality of system sizing and on the time and cost of the acquisition, a user should only modify workload characteristics that will not appreciably alter the values of the performance measures employed in the test.

TABLE OF CONTENTS

CHAPTER 4. PROCEDURAL GUIDANCE FOR BENCHMARKING

TABLE OF CONTENTS CONTINUED

CHAPTER 4. PROCEDURAL GUIDANCE FOR BENCHMARKING

1. Scope.

a. This chapter presents procedural guidance to Federal agencies concerning the four steps in the benchmarking process listed in figure 4-1. These four steps are particularly crucial to the successful use of remote terminal emulation, and the discussions of these steps concentrate on those areas most important to emulation benchmark tests. The benchmarking process includes many other steps not discussed in this handbook; e.g., workload definition and analysis, development of the acquisition strategy, validation of benchmark tests, conducting the LTD. The benchmarking goal levels achieved in an acquisition depend on the manner in which a user conducts all benchmarking steps. A user, therefore, should study and follow all applicable acquisition and benchmarking policies, regulations, procedures, and guidance, including, in particular, FIPS PUB 42-1. Appendix D provides a bibliography of relevant policy and technical materials.

DEVELOPMENT OF THE BENCHMARKING STRATEGY
PREPARATION OF TELEPROCESSING ELEMENTS FOR BENCHMARK TESTS
PREPARATION OF LTD DOCUMENTATION
USER-VENDOR COMMUNICATION

Figure 4-1. Selected benchmarking steps

b. This chapter contains specific technical information that will assist each Federal agency to decide when and how to use remote terminal emulation during competitive ADP system acquisitions. Part 1, Development of the Benchmarking Strategy, discusses in detail several major areas that a user should analyze during the development of the overall benchmark structure; these areas include the types of workload demands in each capacity test, the benchmarking technique(s) used, the teleprocessing (TP) performance measures employed, etc. Several alternatives are presented, along with advantages, disadvantages, and recommendations. Part 2, Preparation of Teleprocessing Elements for Benchmark

Tests, describes two categories of TP elements (scenarios and configurations of TP devices and data communication links) that a user must prepare for each emulation benchmark test. This part outlines the typical actions necessary to design, construct, test, and document these two categories of TP elements, recommends a specific scenario format, and provides an example scenario. Part 3, Preparation of LTD Documentation, discusses the emulation-related documentation that a user should provide to vendors describing how the LTD is to be conducted. This part identifies specific TP elements that must be documented and suggests possible formats for this documentation; example TP elements include the assignment of scenarios to TP devices and data communication links, and the TP performance measures used and the required levels of performance. Part 4, User-Vendor Communication, recommends specific, minimum technical communication between a user and vendors during an acquisition involving remote terminal emulation. This communication is necessary because of the great technical complexity of most emulation benchmark tests, and can greatly reduce benchmark discrepancies, increase competition, and reduce user and vendor time and costs.

c. The emulation and benchmarking capabilities that a user may require vendors to provide during an acquisition are not defined in this chapter. These capabilities are specified in chapter 5. Some of the discussions in this chapter however, assume that the reader is familiar with the emulation specifications in chapter 5.

2. Audience. The objective of this chapter is to present to users procedural guidance concerning when and how to use remote terminal emulation during acquisitions. The primary audience for this chapter, therefore, is composed of the manager of the agency benchmark development project and each agency benchmark analyst. Vendor personnel, however, also will benefit from this chapter, because it will help them understand the benchmarking approaches, terminology, and documentation used by agencies.

3. Admonition. The following admonition is adapted from FIPS PUB 42-1, and applies to the guidance in this chapter:

> The user should keep two basic principles in mind in reading and using this guidance. One is that the guidance is composed of general descriptions of good practices for the normal situation. They do not cover nor are they applicable in all situations. The second

principle is that the guidance stresses reasonableness in all practices and procedures. Reasonableness, in general, is a user determination. The user is solely responsible for determining his organization's requirements, for constructing benchmark tests and an LTD reflecting these requirements, and for ensuring that all decisions made during the entire process are in accordance with all applicable policy, regulations, etc. Any question of procedure or technique should be evaluated in this context and ultimate decisions should protect the Government's interest. The guidance in this section does not contain procedural steps that can be followed as a "recipe" with successful results. Instead, the guidance is simply a discussion of good practices associated with areas of concern. In this sense, the guidance is useful as a checklist and, to some degree, identifies areas where special competence, expertise, or particular attention is indicated.[1]

[1] National Bureau of Standards, "Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment," FIPS PUB 42-1 (Washington, DC: NBS, May 1977), p. 5.

PART 1. DEVELOPMENT OF THE BENCHMARKING STRATEGY

4. General.

a. A user should carefully analyze and define the benchmarking strategy for an acquisition before preparing any of the individual benchmark test elements. A good benchmarking strategy is needed early in the acquisition so that (1) all tests complement each other and achieve the benchmarking goal levels chosen by the user and (2) later benchmarking efforts are properly integrated into the acquisition strategy.

b. This part discusses eight major technical areas a user should analyze during the development of a benchmarking strategy; these areas are listed in figure 4-4. The discussions assume that the user has decided during the development of the acquisition strategy to use a capacity test. It also assumes that the user has estimated and quantified workload requirements throughout the contractual life of the new system; i.e., has completed workload definition and analysis. (Depending on the benchmarking strategy developed, the user may need to gather additional workload data to prepare the elements of the benchmark tests.) The discussions usually are limited to strategic aspects directly affecting remote terminal emulation during capacity tests; most aspects of batch benchmark tests are not addressed.

c. The material in this part complements the guidance presented in FIPS PUB 42-1, primarily Section III.B (Benchmarking Philosophy, pp. 11-12) and Section IV.B (Workload Mix, p. 18). The user, therefore, should also study these sections of FIPS PUB 42-1 before developing a benchmarking strategy.

5. Workload types in capacity tests.

a. A user can employ many valid techniques to determine the workload types to include in capacity test(s). It is beyond the scope of this handbook to discuss all of these techniques, because the specific technique(s) that are employed (1) usually depend on the approaches used to define the workload requirements, and (2) are often independent of whether or not remote terminal emulation is used. (The National Bureau of Standards is preparing a separate guideline on approaches for defining workload requirements, and specific techniques for determining the workload types in benchmark tests.) This handbook does recommend, however, that all users employ at least one fundamental technique during the development of the benchmarking strategy. (Additional techniques should be used during the preparation of each test.) The recommended technique consists of (1) identifying all significant generic types of workload demands projected for the user's operational environment throughout the contractual life of the acquisition, (2) evaluating the probable effect on each benchmarking goal of including each generic type of workload demand in a capacity test, and (3) choosing those generic types of workload demands that result in the combination of goal levels chosen for the acquisition. Generic types of workload demands include both generic applications and generic TP devices. (Different generic applications impose on a SUT different workload demands associated with SUT hardware and system support software; e.g., data base management, remote batch job entry, etc. Similarly, different generic TP devices impose on a SUT different workload demands associated with supporting the characteristics of the device; e.g., character set, line protocol, CRT buffer size.) In this step, the user identifies the generic types of workload demands (workload types) to include in the test(s), but does not determine either the amounts or the detailed characteristics of each type.

WORKLOAD TYPES IN CAPACITY TESTS
BENCHMARKING TECHNIQUES
SCENARIO-WORKLOAD CORRESPONDENCE
TELEPROCESSING PERFORMANCE MEASURES
METHODS FOR REPRESENTING CHANGING WORKLOAD
WORKLOAD SCHEDULING PROCEDURES
BENCHMARK VERIFICATION TECHNIQUES
BENCHMARKING FOR INSTALLATION VERIFICATION

Figure 4-4. Selected technical aspects of benchmarking strategy

b. A user should list all the generic types of workload demands projected to occur during the life of the acquisition and then, if possible, itemize more specific types of workload demands within each generic type. Possible generic applications include interactive program development and testing, remote batch program development and testing, interactive document preparation, interactive data entry, interactive scientific problem solving, bulk data transfer, and interactive data inquiry. Specific types of workload demands for these generic TP applications can include, for example, a particular user-developed application system, a known third-party data base management system, and document preparation utilities to be proposed by the system vendor. Possible generic TP devices include interactive asynchronous teleprinters, interactive asynchronous displays, interactive synchronous teleprinters, interactive synchronous displays, remote batch terminals, remote host systems, cluster controllers, concentrators, and packet network interface devices. Specific types of workload demands for these generic TP devices can include, for example, transmission speeds, formatted screen capability, device make and model, and character sets and protocols; e.g., ASCII, ANSI X3.66-1979 synchronous line protocol.

c. The user should evaluate the probable effects on each benchmarking goal of including in a capacity test each of the itemized generic and specific workload types. All generic and specific types should be ordered by the impact of each type on benchmark representativeness, system sizing, and all other benchmarking goals. The degree to which a workload type impacts representativeness reflects not only the relative volume of that workload type compared to the other types, but also the criticalness of that type to the user's mission requirements; i.e., the loss that would result if the acquired system were unable to accomplish a certain workload type at the necessary level of performance. The degree to which a workload type affects sizing depends, in part, on the workload types needed to size those SUT hardware and software components that the user believes will significantly impact the price and/or performance of the acquired system; e.g., CPU, main memory, channels, front-end processor (FEP), COBOL compiler, text editor, data base management system. The user should evaluate the time and cost to both the user and vendors of including each generic workload type, as well as each specific type, in a capacity test. The amount and transportability of user-provided application source code and the size and transportability of data files affect the time and cost of benchmarking, and should be evaluated. Other cost factors which should be considered are (1) the complexity of the scenarios; (2) the approximate execution time of each workload type; (3) the complexity of including many different workload types in capacity tests; and (4) the user time and cost required to package, document, and distribute the application code and/or data files. The user should know whether the SUT features necessary to support each workload type are mandatory or optional in the RFP. A user should not include workload types that require optional SUT features that some vendors may be unwilling or unable to bid; e.g., support for certain line protocols and make and model terminal types.

d. A user should select for the capacity tests those workload types that result in the combination of benchmarking goal levels chosen for the acquisition. Workload types usually should be included when they significantly increase both the representativeness and quality of system sizing without either increasing the time and cost of the acquisiton or reducing competition. A workload type that has only a minor impact on benchmark representativeness, but that would greatly increase the acquisition time and cost, typically should be omitted from capacity tests. (A workload type that is not important to representativeness but is critical to the user's mission could be omitted from a capacity test but be included in a functional test.) A specific application or TP device could be represented by another of the same generic type. Such a substitution often can reduce the time and cost of the acquisition without significantly decreasing either representativeness or the quality of sizing. Sometimes a substitution may be the only practical alternative to the total omission of a workload type that is important to the user but that cannot be included in a test; e.g., an application that is not yet operational or requires major conversion, or a certain make and model interactive terminal that cannot be emulated by vendor RTE's. Both the estimated price of the system to be acquired and the value and criticalness of the user's mission should be reflected in the variety and complexity of the workload types included in benchmark tests. Moreover, a user typically should omit any workload types when the value of including that type in a capacity test is questionable.

6. Benchmarking techniques. For each benchmark test, the user should select one or more benchmarking techniques that can impose the chosen workload demands on the SUT's. Three benchmarking techniques are recommended for use by Federal agencies in capacity tests: local batch, real TP devices,

and remote terminal emulation.[2] Only local batch and real TP devices (one or two) are recommended for functional tests, however, because such techniques are almost always more cost-effective than remote terminal emulation. The benchmarking technique(s) that a user should employ depends on the types of workload demands chosen for a test and the benchmarking goal levels desired by the user.

a. Local batch. The simplest benchmarking technique is local batch, wherein batch programs and/or data are input and output during benchmark mix execution through SUT peripherals directly connected to the SUT; e.g., card readers, line printers. Occasionally, the batch input queue may be loaded before the start of the execution. Local batch can impose most types of background batch execution demands, and local peripheral I/O demands that are produced by batch activity originating through either local SUT peripherals or remote TP devices; e.g., remote batch terminals (RBT's). The major advantages of this benchmarking technique result directly from its simplicity. Local batch is the least costly and time-consuming benchmarking technique, can be used with almost all vendors, can result in very high benchmark uniformity and repeatability, and usually maximizes competition. Local batch, however, cannot be used to represent the types of workload demands produced by TP overhead processing and data transmission; e.g., protocol handling, queuing, scheduling. Such overhead TP workload demands often are a substantial part of the total workload in systems supporting large numbers of TP devices (especially interactive terminals) and/or a high volume of data transmission. These workload demands also strongly influence the number and the hardware and software characteristics of SUT front-end data communication processors (FEP's). In addition, performance measures determined with benchmark tests which use the local batch technique often are not indicative of the performance that a user would receive at a TP device. For example, the completion time for a data base inquiry submitted by a local

---

[2] Occasionally, agencies have used benchmarking techniques where a vendor-provided driver executes within the SUT; e.g., in a CPU or an FEP. Such internal benchmarking techniques are not recommended for use during Federal ADP system acquisitions, primarily because these techniques (a) vary greatly in the types of workload demands that can be represented, (b) are not available from all system vendors, (c) greatly reduce the quality of system sizing, and (d) increase the likelihood of benchmark discrepancies.

batch job can be significantly different from the response time for the same inquiry submitted from an interactive terminal, because the batch completion time does not include common delays due to polling, FEP processing, inquiry scheduling, queuing, etc. For most TP systems, therefore, local batch usually results in low benchmark representativeness and poor quality system sizings.

b. Real TP devices.

(1) A common benchmarking technique uses one or more real TP devices to impose TP workload demands on the SUT. This document defines two broad categories of TP devices. One category, referred to as remote devices, is composed of those TP devices where user workload demands originate. Sample remote devices include interactive teleprinters, remote batch terminals, and remote host systems. The other category, referred to as intermediate devices, is composed of those TP devices used to connect remote devices to a host computer system. When used, intermediate devices are configured between remote devices and host systems. Sample intermediate devices include terminal cluster controllers and concentrators. When TP devices are installed for a benchmark test, they may be part of a vendor's proposed configuration and, therefore, may be under evaluation. TP devices, however, may be extra components configured only to test the SUT. Remote devices usually are operated by vendor personnel who follow dialogues developed by each vendor from user-provided scenarios.

(2) Real TP devices can impose on a SUT all the types of TP workload demands that can arise in the user's operational environment, and, therefore, potentially can maximize benchmark representativeness. The degree to which high representativeness and other benchmarking goals can be achieved, however, depends primarily on the number of remote devices needed for a specific benchmark test. A test involving more than a few remote devices is impractical because of (a) the difficulty of training and coordinating the personnel to operate the remote devices; (b) the low benchmark uniformity and repeatability that results from the inevitable variability and errors of human operators; and (c) the often monumental time, cost, and physical space required to assemble remote devices, communication lines, and operators for the many benchmark mix executions needed prior to and during an LTD. Tests using less than five remote devices, however, can be conducted by most vendors without large investments of time and funds and within acceptable levels of uniformity

and repeatability. Unless a SUT has little TP capacity, however, five remote devices alone cannot produce the volume of TP workload demands usually needed for high benchmark representativeness and high quality system sizing. Real TP devices, therefore, should not be used alone for capacity tests of SUT's intended to support large volumes of TP workload demands. Real TP devices are well-suited for functional tests, because these tests almost always can be conducted with one or two remote devices and device operators.

(3) The use of real TP devices for benchmark tests affects both the acquisition time and costs for vendors and the probable level of competition. The magnitude of the effect depends not only on the number of devices needed for a test, but also on whether or not the vendors bid the TP devices as part of their proposals. When a vendor bids TP devices as part of its proposal, that vendor usually is willing to install and operate one or two of each specific make and model device bid. When a vendor does not bid TP devices, however, the vendor often does not have, and is reluctant to acquire, the make and model TP devices desired by the user. This is particularly true when a device is not part of a vendor's product line. When a user decides to employ real TP devices that are not bid by a vendor, the user should either (a) permit the vendor to substitute TP devices of the same generic types as the specific devices desired or (b) provide the specific devices to the vendor for use during the preparation and conduct of tests.

(4) When a Federal agency employs real TP devices for a benchmark test, the agency shall not require a vendor to physically install and operate for a single test more than five remote devices of any combination of types and characteristics. When a vendor bids the TP devices, the agency may require the vendor to install and operate up to two TP devices of each specific make and model bid, provided that the total number of remote devices is not more than five. When a vendor does not bid the TP devices, the agency may require the vendor to install and operate up to two remote devices of the same generic type as the device desired, provided that the total number of remote devices is not more than five.

c. <u>Remote terminal emulation</u>.

(1) Remote terminal emulation, like real TP devices, can be used to impose on a SUT all the types of TP workload demands that can occur in a user's operational environment. In practice, the specific workload types that

can be imposed depend upon the remote terminal emulation capabilities available for a test. Chapter 5 defines the remote terminal emulation capabilities that (a) Government agencies are permitted to require vendors to provide for capacity tests conducted at vendors' facilities during TP system acquisitions and (b) should be common among all computer system vendors participating in Federal TP system acquisitions. A user should study these specifications before deciding whether or not to use remote terminal emulation during a specific acquisition.

(2) The primary advantage of remote terminal emulation over local batch and real TP devices is emulation's ability to impose on a SUT high volumes of TP workload demands, including the workload demands produced by TP software utilities, TP overhead processing, and data transmission. For a SUT designed to support large numbers of TP devices and/or a high volume of data transmission, remote terminal emulation is the only practical benchmarking technique for achieving high benchmark representativeness and high quality system sizing. Vendor time and cost are usually lower with emulation than with real TP devices when more than five remote devices and device operators are used in a single test. When vendors and users follow the emulation guidance and specifications contained in this handbook, high benchmark repeatability and uniformity can be achieved and benchmark discrepancies can be minimized. Well-defined performance measures are available with emulation for equitably comparing SUT performance.

(3) The primary disadvantages of remote terminal emulation are based on the limited availability of RTE's and the high vendor costs for emulation benchmark tests. While almost all manufacturers of medium and large TP systems have RTE's and extensive benchmark test facilities, some minicomputer and plug-compatible mainframe vendors have little or no benchmarking or emulation capabilities. The mandatory use of remote terminal emulation, therefore, may reduce the level of competition in some acquisitions, particularly for minicomputers. Another disadvantage is that RTE's typically cannot be used at a user site, because (a) most vendors do not distribute RTE software and (b) most user sites do not have sufficient hardware to conduct an emulation benchmark test. A user, therefore, typically cannot execute the benchmark mix, including emulated TP devices, during final preparation and integration of the benchmark mix. For the same reasons, an emulation benchmark test cannot be repeated at a user site for installation verification. Vendor time and cost to conduct an emulation benchmark test is usually

higher than for a test using only local batch or a few real TP devices, and reflect the personnel hours, the SUT and RTE hardware costs, and machine time needed to prepare each script and to execute repeatedly the total mix during system sizing. A vendor with emulation capabilities, therefore, may decide not to compete in an acquisition if the probable value of the resulting contract is too small to justify the vendor's time and cost to conduct emulation benchmark tests. User time and cost to design, prepare, and document emulation benchmark tests typically are also higher than for other tests, and reflect the effort needed to develop scenarios and procedural documentation. (User time and cost to develop a single scenario is about the same whether real TP devices or an RTE is used; more scenarios, however, are typically developed when an RTE is used.)

(4) Each vendor's RTE primarily emulates only TP devices in that vendor's product line. It is impractical for an RTE to emulate correctly all the specific make and model TP devices that all users desire to represent for test purposes, because of the enormous number of different devices. Therefore, when a vendor bids TP devices, a user can expect the vendor to emulate the specific make, model, and operational characteristics of the devices bid. When a vendor does not bid TP devices, however, a user must allow a vendor to emulate a TP device of the same generic type as the specific make and model device desired by the user. Chapter 5 defines the generic device types and characteristics that a user may require vendors to emulate.

(5) Each Federal agency shall choose for itself whether or not to use remote terminal emulation during a TP system acquisition. The choice necessarily depends upon the specific circumstance of each acquisition and primarily reflects the user's judgements about:

(a) The criticalness of the user's ADP mission requirements; e.g., the potential loss if the acquired system is unable to satisfy user requirements;

(b) The types and volumes of TP workload demands;

(c) The estimated total dollar value of the resulting contract;

(d) The increased benchmark representativeness and quality of system sizing that would result from using emulation, and the value of the increases to the user;

(e) The amount and availability of user personnel, time, and funds needed to use emulation;

(f) Possible disadvantages due to increased vendor marketing costs and the potential reduction in competition; and

(g) The levels of mission and cost risks acceptable to the user.

(6) It is recommended that remote terminal emulation not be used for any functional tests. Except under extraordinary circumstances, such as extreme mission criticalness, remote terminal emulation also is not recommended for any capacity test when one or more of the conditions listed below apply. In all other circumstances, no general recommendations can be given.

(a) The estimated total amount of the resulting contract is not more than $400,000. The estimated total includes the costs of all hardware, software, maintenance, etc. for the initial and all optional years of the contract.

(b) A total of no more than five remote devices would be emulated for the capacity test.

(c) The total estimated TP workload on the SUT, as quantified by such indicators as the total CPU time or disk I/O's attributable to TP workload demands is less than 15 percent of the user's total workload; sample TP workload demands include TP overhead processing, TP utilities, and TP applications.

(7) When a Federal agency chooses to use remote terminal emulation during its TP system acquisition, the agency shall follow all mandatory, RTE-related acquisition policies, regulations, procedures, and guidance in effect at the time of the acquisition. (An agency shall contact GSA if, due to extraordinary circumstances, it desires to deviate from these mandatory, policies, regulations, procedures, or guidance.) It is mandatory that all RTE's used during LTD's will be provided and operated by the vendors participating in the acquisition. The agency may require vendors to provide any combination of the emulation capabilities defined in this handbook, provided the agency has determined that the emulation capabilities are needed to determine the capacity of the SUT hardware and software components actually bid by each vendor. Therefore, for a particular acquisition,

an agency may require all vendors to provide the emulation capabilities needed to determine the capacity of the mandatory support items. For desirable (optional) support items, an agency shall require only those vendors who bid the desirable items to provide the emulation capabilities used to determine the capacity of the desirable items. Regardless of the mandatory and desirable SUT support items, however, an agency shall not require emulation capabilities that are not explicitly defined in the handbook. An agency shall also not require vendors to provide any emulation capabilities for benchmark tests conducted at the agency's site; e.g., a pilot test, installation verification. (An agency shall obtain a waiver from GSA if the agency desires to require vendors to conduct emulation benchmark tests at the agency's site for a major systems acquisition as defined by Office of Management and Budget Circular A-109.) When the benchmark instructions are released to industry, moreover, an agency shall define clearly the emulation capabilities that vendors must provide. An agency is permitted to declare a vendor nonresponsive, and to disqualify the vendor from the acquisition, if the vendor does not provide the subset of the emulation capabilities specified in the handbook that the agency determines is necessary to test the SUT. In all cases, a vendor still retains the right to request from the agency a waiver of any benchmark test and/or remote terminal emulation requirement. The agency retains the right to grant such waivers. (Agencies, however, should carefully evaluate the effect of a waiver on benchmark representativeness and uniformity, as well as on the level of competition, before granting the waiver.)

d. Combination of techniques. For some acquisitions, a combination of two or three benchmarking techniques may be needed in a single capacity test to achieve the desired benchmarking goal levels. When remote terminal emulation is used, it is recommended that, for each device type emulated, one real remote device be installed and operated, up to a total of five real remote devices. Real remote devices can help verify many aspects of the test, such as dialogues and values of some performance measures. Therefore, such devices can help minimize benchmark discrepancies. Whenever such a level of representativeness is appropriate, a user also can employ local batch to represent background batch processing and simultaneously can use real remote devices, and perhaps emulation, to represent TP workload demands. Agencies are urged strongly, however, to choose a level of complexity for each capacity test (especially for tests using remote terminal emulation) that is appropriate to the criticalness of the mission and the probable dollar value of the resulting contract. The level of complexity depends not only on the benchmarking techniques used, but also on such factors as the number and languages of batch jobs, the size and interrelationships of test data, the number, length, and logical complexity of scenarios, the number of emulated devices, and the number of data communication links installed.

7. Scenario-workload correspondence

a. A scenario is a system- and vendor-independent description of a group of TP workload demands to be performed during a benchmark mix execution. A scenario usually should correspond to the primary, self-contained unit of work for each generic type of TP workload demand in the capacity tests; e.g., a transaction or a document retrieval and edit. Many different scenarios typically are used in a single benchmark test. The scenario is the primary unit by which a user describes the types and volumes of TP workload demands in a test and expresses most TP execution procedures and some required performance levels. For a user, therefore, the scenario is the principal TP scheduling and reporting unit for emulation benchmark tests. This handbook defines extensive RTE capabilities for scheduling and reporting both individual and groups of scenarios. (An RTE actually executes scripts, not scenarios. Each vendor produces one or more scripts to perform the workload demands contained in a single scenario. The script language and structure is the responsibility of each vendor and differs from vendor to vendor. The user, in turn, is responsible for verifying that the scripts correctly impose the workload demands described in the scenarios, and that all relevant test procedures have been followed during the translation of scenarios to scripts.)

b. A scenario can consist of almost any arbitrary group of TP workload demands. For example, a single scenario can consist of (1) a complete timesharing user session, beginning with a logon, ending with a logoff, and lasting over 30 minutes; (2) a single, short interactive function (e.g., document retrieval and edit, data base inquiry) that does not include either logon or logoff, that assumes the SUT and interactive terminal are ready for input when the scenario starts, and that returns the SUT and terminal to the same state when the scenario completes; (3) a series of input transactions or remote batch printouts that, once started, can only be ended from the RTE operator console; and (4) a logoff sequence, followed by a random delay and a logon sequence, included in a benchmark mix so that a representative number of logon and logoffs occur throughout the mix execution. In addition, the correspondence between a user's TP workload demands and the scope of each scenario (the scenario-workload correspondence) can be different for each type of workload demand.

c. The scenario-workload correspondence can significantly affect the benchmarking goal levels obtained in a benchmark test, especially the time and cost of benchmarking, the quality of system sizing, and the benchmark representativeness. A user, therefore, should carefully analyze and select the scenario-workload correspondence for each type of TP workload demand included in an emulation benchmark test. It also is recommended, however, that scenarios be as short and as simple as possible, and not include any alternate logical paths, logic checks, etc., because such complexity usually increases both vendor time and cost to conduct the tests and the chance of benchmark discrepancies. Instead of alternate operations within a single scenario, scenario scheduling rules should be used to control the types, volumes, and distribution of TP workload demands in a test. In general, a user should attempt to design each scenario so that it is a simple sequence of inputs, outputs, and delays.

8. Teleprocessing performance measures.

a. During the development of a benchmarking strategy, a user must choose the performance measure(s) that he will use to define the performance levels required for each capacity test. In this step, a user selects only the performance measures, not the required value(s) for each measure. The handbook describes in detail the TP performance measures and summary reports that a user can require from vendor RTE's. A user should study these before selecting TP performance measures.

b. Throughput, the most basic TP performance measure used during a capacity test, is defined in this handbook as the number of user workload demands successfully completed within a predefined time. To use throughput, an agency must specify (1) the exact types of workload demands to be completed, (2) the exact (or minimum) number of each type of workload demand to be completed, and (3) the maximum elapsed time allowed for the SUT to successfully complete this number of workload demands. Throughput often is used to define the required performance level for completing batch programs. For example, the proposed system must successfully complete one execution of Program A, four executions of

Program B, and two executions of Program C within an elapsed time no greater than 36 minutes. For TP workload demands, throughput can be used to define the required performance level for completing scenarios. A user may specify throughput requirements for each individual scenario or for user-defined scenario groups.[3] For example, within 62 minutes the proposed system must complete successfully 22 executions of Scenario A and 16 executions of Scenario B; within 50 minutes, the system must complete successfully at least 90 scenarios from Scenario Group I and at least 65 scenarios from Scenario Group II. Throughput can also be used to define the required performance levels for completing from one to 20 different vendor-independent user functions. A user function is a single, logically related user work item included in a scenario, and each vendor may require a different number of operator inputs and SUT outputs to accomplish the same user function; e.g., replace all occurrences of "estimated cost" in a document with "estimated price" and, after the replacement, display the changed sentences; complete an order-entry transaction. For TP workload demands, a user also may employ throughput with a single pair of data transmissions exchanged by an emulated TP device and a SUT; e.g., a single timesharing command and the resulting SUT output, a one-line inquiry and the resulting SUT response. Such application-related exchanges of data transmissions are called application I/O pairs and vary greatly from vendor to vendor and from SUT to SUT. (See chapter 2, part 2.) In many cases, a user may not be able to identify uniquely an application I/O pair that can be used for comparing the performance of different SUT's and, therefore, may not be able to use throughput with application I/O pairs.

c. Turnaround time is an important TP performance measure used during emulation benchmark tests and is the time interval between the initiation of a user workload demand and the successful completion of that workload demand. To use turnaround time to define the performance levels required for completing workload demands, an agency must specify precisely the workload demands to be completed and

[3] Chapter 5 specifies emulation capabilities for the definition, scheduling, and performance analysis of up to 20 different scenario groups. A user may prefer to treat several different scenarios as a group, especially when all the scenarios in a group correspond to variations of the same generic workload type; e.g., Scenario Group I corresponds to interactive document preparation and includes three different scenarios in certain proportions.

some statistical description of the elapsed time allowed for the SUT to successfully complete each occurrence of each workload demand. Turnaround time can be used with scenarios, scenario groups, and user functions. Chapter 5 specifies that the following turnaround time statistics be available in RTE log summary reports: average, minimum, maximum, median, and up to four optional, user-defined percentile levels. The following are examples of how an agency can use turnaround time to define the required TP performance levels:

(1) For all successfully completed executions of Scenario A, the average turnaround time must not exceed 14 minutes, and 95 percent of all the turnaround times must not exceed 16½ minutes.

(2) For all successfully completed executions of all scenarios in Scenario Group I (document preparation), the average turnaround time must not exceed 20 minutes, 80 percent of the turnaround times must not exceed 22 minutes, and no turnaround time may exceed 25 minutes.

(3) For all successfully completed occurrences of User Function 1 (order entry transaction), 60 percent of the turnaround times must not exceed 4 minutes, and 98 percent may not exceed 6 minutes.

d. Response time is another TP performance measure sometimes used during capacity tests. Response time, defined only for interactive workload demands, is defined in this handbook as the elapsed time from the last keystroke of an operator input at an interactive remote device until the first printable character of the resulting SUT response appears at the user's device.[4] To use response time during emulation benchmark tests, an agency must (1) identify precisely one or more interactive workload demands so that, for every SUT of every participating vendor, each demand is accomplished by a single application I/O pair; (2) determine, for every SUT of every participating vendor, that the first

---

[4] This basic definition of response time, developed to be technically valid, consistent, and practical for all vendors during emulation benchmark tests, is to be used by all RTE's that satisfy the emulation specifications contained in this handbook. Agencies shall use this definition with emulation benchmark tests during system acquisitions, even though this definition differs from the definition of response time contained in FIPS PUB 57, "Guidelines for the Measurement of Interactive Computer Service Turnaround Time and Response Time."

printable character of the resulting SUT output actually indicates that the SUT has completed the workload demand; and (3) specify a statistical description of the response times allowed for the SUT to successfully complete each occurence of each workload demand. Response time can only be used with application I/O pairs. The following response time statistics are available in RTE summary reports: average, minimum, maximum, median, and up to four optional, user-defined percentile levels.

e. A user should carefully evaluate each TP performance measure before selecting the measure(s) to be employed in a capacity test, because the measures employed are critical to the benchmarking goal levels achieved. When several different generic types of workload demands are included in a single benchmark test, a user may decide to use a combination of performance measures. Increasing the number of performance measures in a test, however, also increases (1) the complexity of the test, (2) the chance of benchmark discrepancies, and (3) the difficulty, time, and cost for vendors to achieve high quality sizing. In addition, increasing the number of measures may not significantly improve benchmark representativeness. Therefore, a user typically should attempt to minimize the number of different measures employed. It is recommended that in each capacity test agencies use throughput as the fundamental TP performance measure. Throughput is the most vendor- and system-independent measure, and often is the most mission-oriented measure as well. The use of throughput also encourages high benchmark representativeness and uniformity and reduced benchmark discrepancies. In addition to batch jobs, throughput typically should be employed with scenarios and/or scenario groups, but should not be used with application I/O pairs. When an agency decides to use other TP performance measures together with throughput, it is further recommended that the agency choose turnaround time, and use it with scenarios, scenario groups, and/or user functions. The required values of turnaround time should be defined in terms of averages and/or percentiles, not as absolute permissible values.

f. In emulation benchmark tests involving up to several hundred emulated TP devices and data communication links, transient hardware and software problems sometimes occur that can cause a small number of emulated devices and/or a small number of scenarios to abort. In many previous acquisitions, these tests were declared invalid and had to be repeated. This increases benchmarking time and cost. When the failures are statistically small, such as one scenario failure out of 200, the value to either the vendor or the agency of repeating the test is questionable. Agencies should evaluate the significance of such benchmark discrepancies before a test is declared invalid and repeated. In capacity tests using throughput as a performance measure, one possible approach for accommodating such a small number of failures is for the agency to (1) define a minimum number of each type of workload demand that must be completed successfully, instead of an exact number, and (2) permit each vendor to configure more links, emulate more devices, and complete more scenarios than the minimum required so that a minor failure would still allow the vendor to meet the required performance level and, thus, avoid a repetition of the test. Each vendor would be responsible for choosing the numbers and types of "extra" workload demands in such a test, based on the vendor's experiences and, perhaps, limited by an agency-defined maximum; e.g., 2 percent or 5 percent. This approach is only feasible, however, when an agency provides sufficient procedures and benchmark test elements, such as data files and transactions, so that additional workload demands can be added to the benchmark mix.

g. It is recommended that agencies not use response time with capacity tests during TP system acquisitions. The relevance and technical validity of comparing two TP systems by their response times decreases as the differences increase between the dialogues and application I/O pairs used to complete identical workload demands. The most critical differences are (1) the number and complexity of operator inputs, and (2) the significance of the first printable character of the SUT output resulting from each operator input. It is impractical for an agency to mandate and achieve identical numbers and complexities of operator inputs for all vendors, except for agency-supplied TP applications where the dialogue and application I/O pairs are identical for all SUT's. Each vendor, therefore, develops a dialogue for each scenario that will lead to the best values for the performance measures used in the test. The use of response time to define the performance levels required in a capacity test, therefore, often lowers benchmark representativeness and uniformity and the quality of system sizing.

h. For emulation benchmark tests, it is recommended that agencies use only the TP performance measures and performance reports described in this handbook. The measures are well-defined and technically consistent for all vendors, and the reports provide sufficient information and flexibility to compare the performance of almost all TP systems. The use of these measures and reports also will reduce benchmarking time, cost, and discrepancies for users and vendors. If an agency desires additional TP performance measures and/or reports, the agency shall prepare any and all RTE log reduction and report generation programs needed to produce the additional reports. It is mandatory that all agency-prepared log analysis programs be in an ANSI standard language (e.g., COBOL), use the RTE Log Summary Tape (described in chapter 5, part 6) as input, and be fully described and distributed to vendors when the benchmark instructions are released.

9. Methods for representing changing workload.

a. A user typically prepares capacity tests to size the proposed system at several points throughout the contractual life of the acquisition. This requires the user to design and prepare benchmark test elements that reflect projected changes in the types and/or the volumes of workload demands. A user typically employs one of three basic methods to reflect in benchmark tests the changing workload: (1) Use the same benchmark mix with different (typically better) required levels of performance, (2) use different benchmark mixes with different performance levels, or (3) use different benchmark mixes with the same required performance levels.

b. The representation of changing workload demands by using different required performance levels with the same benchmark mix is based on the following assumption: If a SUT can execute some Volume V of workload demands within some performance Level L, then that SUT has the same capacity as another SUT that can execute n times Volume V within n times performance Level L. For example, a system that can complete a mix containing 100 batch jobs within 60 minutes is assumed to have the same capacity as an another system that can execute a mix of 50 jobs within 30 minutes. (The two mixes also must contain the same proportions of each type of workload demands.) This assumption permits a user, for example, to represent a doubling of workload demands by reducing by half the permitted completion time for the same mix, instead of by doubling the volume of demands in the mix. This approach reduces a user's time and cost to prepare different mixes and also reduces vendors' time and costs to conduct benchmark tests. A user readily can reflect fractional workload changes, because the numbers used to express performance levels (minutes) can be adjusted in finer precision than most units of workload; e.g., batch jobs or interactive sessions. This assumption is not valid, however, for all types of workload demands and all performance measures, especially for workload types and measures that incorporate operator actions and delays; e.g., tape mounts, think time. It also is usually true that much more than twice as much capacity is required to reduce response time by half. This approach also cannot represent the additional TP overhead needed to support more simultaneously active remote devices, nor can it represent changes in the types of workload demands. For TP systems, this approach often results in reduced benchmark representativeness and poorer quality system sizing than other approaches for reflecting workload changes.

c. The representation of changing workload demands by using different benchmark mixes with different required performance levels provides the user the greatest flexibility to design and prepare benchmark tests. A user can adjust the contents of each mix and each required performance level to maximize benchmark representativeness and the quality of sizing. For example, TP workload demands may be changed to reflect different numbers and types of concurrent remote devices and TP demands, but batch demands may remain constant. The required minimum scenario turnaround time may remain constant, reflecting constant user needs, while the maximum allowable time for completing the batch demands may be reduced. The primary disadvantage of this approach for representing workload change is the increased complexity of the benchmark tests and, therefore, greater user and vendor time and cost to design, prepare, document, and/or conduct the tests. Changes to the required performance levels also may depend on the assumption described above, and, thus, actually may result in reduced representativeness.

d. The representation of changing workload demands by using different benchmark mixes with the same required performance levels allows the user the flexibility to modify the types and volumes of workload demands in a mix without the additional complexity of simultaneously adjusting some or all performance levels. For most TP acquisitions, this approach can be used to achieve an appropriate level of benchmark representativeness and quality of sizing. Most TP systems, in fact, grow by adding concurrent users who desire constant levels of performance, not by reducing think time, type time, or required performance levels. The principal disadvantages of this approach, compared to other approaches, are (1) the additional user time and costs that may be needed to prepare and test a greater number of different benchmark mix elements, particularly batch elements, and (2) the reduced ability to reflect fractional workload changes, which results from the granularity of the user work items; e.g., 5 percent growth in 30 active terminals. These disadvantages, however, can be fewer than the advantages obtained through reduced test complexity and increased benchmark representativeness and quality of sizing.

10. Workload scheduling procedures.

a. During the development of a benchmarking strategy, a user should carefully analyze and select the general procedures (if any) to be used by the vendor to schedule workload demands during the conduct of each capacity test. Workload scheduling procedures can impact significantly the benchmarking goal levels achieved during an acquisition, especially benchmark representativeness and uniformity, system sizing quality, and acquisition time and cost. Because agencies have some control over the scheduling of batch workload demands in most operational systems, they typically use few workload scheduling procedures in capacity tests of batch systems and, therefore, allow vendors reasonable flexibility to demonstrate SUT efficiencies by scheduling. In operational TP systems, however, agencies often have little or no control over the scheduling of TP workload demands from remote devices. In capacity tests of TP systems, therefore, agencies may need to use several procedures for scheduling scenarios, reflecting the arrival rates, sequences, distributions, etc., of TP workload demands projected in the user's operational environment. Most scenario scheduling procedures are grouped into two classes: (1) Procedures describing how TP workload demands are scheduled at the start of the mix execution (the starting conditions) and (2) procedures describing how TP workload demands are scheduled during the timed portion of the mix execution. Chapter 5 defines several RTE features for implementing various scenario scheduling procedures.

b. Scenario scheduling procedures describing the starting conditions for a capacity test can reduce several potential benchmarking problems. One such problem is a highly unusual distribution of TP workload demands at the start of the timed portion; for instance, 70 interactive terminals attempting to logon within the first minute of the mix execution or all emulated remote devices issuing a file access command almost simultaneously. Another potential problem is lock-step, which occurs when several emulated TP devices that are simultaneously using the same scenario perform the same action at the same time. Lock-step can occur because identical scenarios (1) are scheduled to start simultaneously and/or (2) meet long delays at a common point (such as requesting file I/O), which allows later scenarios to "catch-up" to earlier ones.[5] The long elapsed time sometimes required at the start of a test to achieve the desired distribution of types and volumes of TP workload demands (the long benchmark head) can decrease benchmark representativeness and increase benchmarking time and costs; this potential problem can be reduced by sophisticated starting conditions.

c. Several scenario scheduling procedures have been used to describe starting conditions that reduce some or all of these potential benchmarking problems. One procedure is to allow vendors to choose the starting condition best suited for each SUT; this approach, however, can reduce benchmark uniformity and representativeness, as well as the quality of sizing. Another procedure is to require some number of scenarios to logon prior to the start of the timed portion of the mix execution; this approach alone will not significantly reduce lock-step or the unusual distribution of TP workload demands at start-up. Another procedure is to require all identical scenarios to be staggered at the start of the timed portion of the test. For example, if eight emulated TP devices are scheduled to start a test using Scenario A, the user can divide the scenario into seven parts of approximately equal duration and require the scenarios to be staggered on the emulated devices as illustrated in figure 4-10. The staggering can be performed using the suspend and restart features described in chapter 5, part 4. These features also provide the user time to examine selected emulated devices to verify the staggering before the start of the timed portion of the test. The user can raise or lower the required number of scenarios to be completed during the timed interval, depending on whether or not the user includes the scenarios begun before the timed interval.

d. A user should select procedures to describe how scenarios are scheduled during the timed portion of the test that (1) reflect the arrival rates, sequences, distributions, etc. projected to occur in the user's operational environment; (2) reduce the likelihood of lock-step; and (3) shorten the

---

[5] In addition to scheduling procedures, techniques for reducing lock-step include random think-times and enough unique scenarios so that no more than 10 percent to 15 percent of the active TP devices are ever using identical scenarios simultaneously.

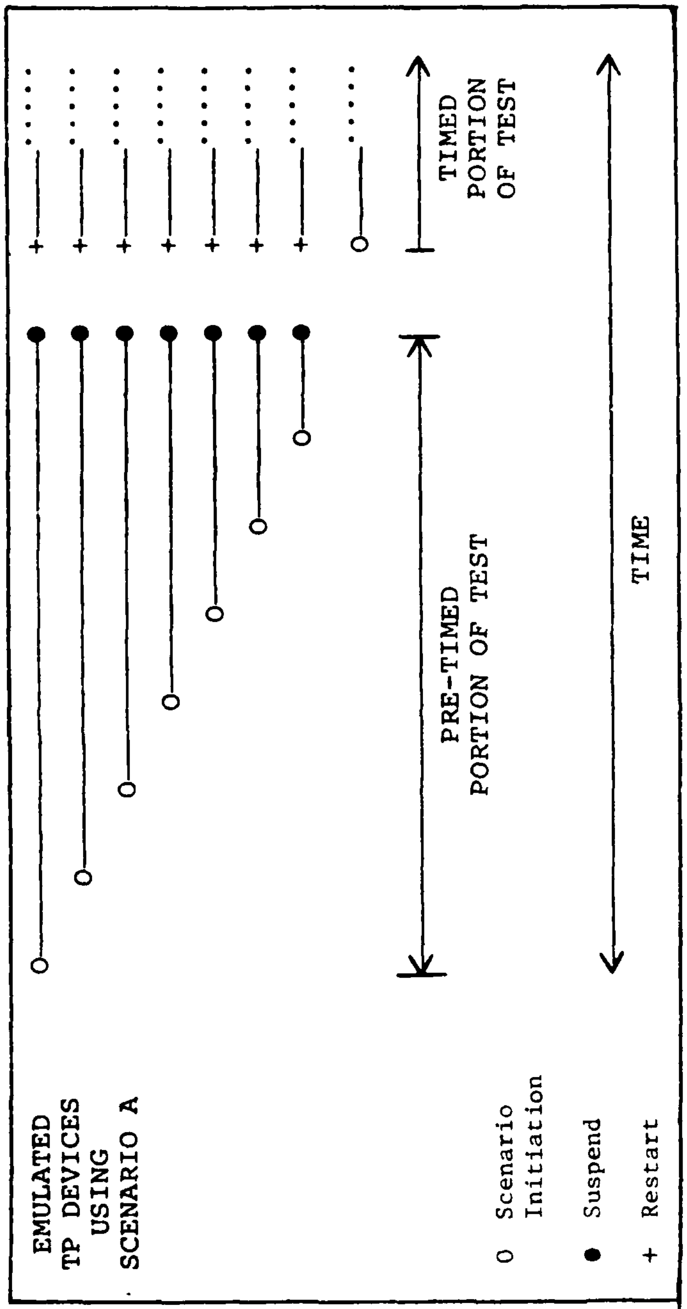


Figure 4-10. Staggered starting conditions using RTE suspend and restart features

benchmark head. One procedure which can reduce the likelihood of lock-step and shorten the benchmark head is to allow vendors to schedule all scenarios so that at least one scenario is active at every point during X percent of the timed interval; however, this technique can reduce benchmark representativeness and uniformity. Another procedure is for the user (1) to preassign every execution of every scenario on every TP device and, perhaps, (2) to permit vendors to adjust the delay time between the end of one scenario on a device and the start of the next scenario on that device (the interscenario delay), so that each emulated device is active X percent of the timed duration. This approach is straightforward and increases benchmark uniformity, but it requires more user effort and unknowingly may favor one SUT over another. Another approach is initiation control, described in detail in chapter 5, part 4. Initiation control uses the total history of all scenarios started during a benchmark mix execution to keep the cumulative percentage of each scenario initiated as close as possible to any agency-specified percentage; e.g., 14 percent Scenario A. This technique maximizes benchmark uniformity without favoring any SUT. When using initiation control, a user also may permit vendors to adjust interscenario delays to spread TP workload demands throughout the timed portion of the test.

11. Benchmark verification techniques.

a. General.

(1) To minimize benchmark discrepancies, a user must develop a verification strategy that includes (a) the specific technical and procedural aspects of each benchmark test that will be verified, (b) the magnitude of discrepancies that will be allowed before the test execution is declared invalid, and (c) the verification techniques to be used and the thoroughness of the application of each technique. The use of remote terminal emulation increases the potential for discrepancies, primarily because it increases benchmark test complexity. For emulation benchmark tests, moreover, a user's verification strategy also should include techniques for verifying that each vendor's RTE meets the emulation specifications required by the agency, because vendor compliance with the specifications in this handbook will not be certified by a central Government group.

(2) The verification techniques used during an emulation benchmark test can greatly affect the benchmarking goal levels. For example, some techniques can significantly

increase acquisition time and costs for the user and vendors, and other techniques can reduce benchmark representativeness. Moreover, the set of verification techniques used in a single test sometimes increases the likelihood of benchmark discrepancies, because of the increased complexity of the test procedures required by the techniques. Therefore, a user must carefully evaluate both the individual and collective effects of the techniques on all benchmarking goals before finalizing the verification strategy. It is recommended that a user employ the simplest verification techniques necessary to be confident that the probability and magnitude of discrepancies do not invalidate the test. A user should omit any verification technique of questionable value.

(3) Summarized below are the principal techniques for verifying the TP aspects of emulation benchmark tests. These verification techniques are grouped according to the time period during which each is applied: (a) During benchmark mix preparation, (b) prior to mix execution, (c) during mix execution, and (d) after mix execution. Some of these techniques depend on emulation capabilities described in chapter 3. All of the verification techniques presented below should not be employed in a single test. During the development of the benchmarking strategy, a user should carefully evaluate these techniques and select the best combination for that user's acquisition. (The selected techniques often will be used with other techniques that are needed to verify other aspects of the test; e.g., inspection of batch program source code and SUT hardware.) In the LTD documentation given to vendors, a user should describe clearly all verification techniques to be employed during the acquisition.

b. <u>During benchmark mix preparation</u>.

(1) <u>Date and time of day</u>. A user can define scenarios that access the RTE system clock and use the date and/or time of day values within input messages; e.g., an edit command. Scenarios that request the current time of day from the SUT can also be defined. By employing one or both of these features, and by resetting the values of the RTE and SUT system clocks to different, arbitrary values immediately before the start of a mix execution, a user can

cause data to be stored in SUT data files and/or the RTE log file(s) that will indicate whether the scenarios were completed during the mix execution.[6]

(2) RTE log messages. A user can define scenarios that send a message of up to 40 characters to the RTE log file. The log file will also include the time of day and the identifier of the device using the scenario. A user can use RTE log messages to note significant events during a mix execution; e.g., start and end of each scenario. After the mix execution, an RTE log report can be produced that includes only these RTE log messages. This report can be used to verify the number, sequence, types, etc., of scenarios initiated and completed during a mix execution.[7]

(3) Controlled variability. An important verification technique is to include limited, controlled variability in the benchmark test design. A user can define think time values and interscenario delays by specifying the following basic statistical distributions: truncated negative exponential, truncated gaussian, and uniform. A user can define scenarios that select, in a uniformly random order, entries from tables of character strings and that incorporate the selected strings in input messages. The random number seed used to produce all statistical distributions can be specified

---

[6] A similar verification technique is to require the vendor to broadcast a user-defined message from the SUT to all active terminals (real and emulated) at a user-chosen time during the mix execution. This technique is not recommended because (a) the same degree of verification can be obtained using the date and time of day, (b) not all SUT's and application systems provide such broadcast capabilities, and (c) an unexpected data transmission could be interpreted incorrectly by an RTE as the response to the previous input transmission, causing transient or irrecoverable emulation errors during the mix execution.

[7] A related verification technique is to require each scenario to send one or more messages to the RTE operator console. This technique is not recommended because these messages often (a) overload the RTE console, (b) cause severe performance degradation in the RTE, (c) result in incorrect think time, interscenario delays, etc., and/or (d) appear too rapidly for a user to employ effectively for verification.

by the user immediately prior to each benchmark mix execution. A user can also specify that scenarios be initiated using the initiation control scheduling technique, which dynamically schedules scenarios to emulated devices according to both user-specified percentages and the history of scenario initiations during a given mix execution. When one or more of these capabilities is properly employed to create controlled variability in a test design, a user can greatly reduce the likelihood that a vendor has taken advantage of certain characteristics of the benchmark test to improperly improve performance. (These capabilities can also increase benchmark representativeness by including in the test any statistical variability that naturally occurs in operational environments.) When employing these capabilities, however, a user must be careful to ensure that appropriate levels of benchmark uniformity and repeatability also are achieved.

c. Prior to benchmark mix execution.

(1) Agency-vendor communication. The most important method of minimizing benchmark discrepancies is the exchange of technical, benchmarking information between a user and vendor. Because of the complexity of many emulation benchmark tests, agency-vendor communication is more critical in acquisitions using remote terminal emulation than for other acquisitions. Agency-vendor communication is discussed in detail in chapter 4, part 4.

(2) Review dialogues. At the start of an LTD, a user can obtain and review copies of the dialogues prepared by the vendor from the scenarios. A user can decide to inspect all or only selected dialogues. The inspection can reduce the likelihood that the vendor has (a) misunderstood a scenario or benchmark test procedure and (b) developed a dialogue that relies on certain characteristics of a scenario to improperly improve SUT performance; e.g., execution of a series of interactive commands stored on the SUT.

(3) Review data communication interface. At the start of an LTD and/or before each benchmark mix execution, a user can review the hardware and software elements of the data communication interface between the SUT and the RTE(s) and between the SUT and any real TP devices. The review can consist of both technical discussions and physical inspection of hardware and software. Elements that might be reviewed include, for example, the number of physical data communication links between the SUT and RTE, the speeds and clocking

components for the links, the link protocol used by each link, the FEP software in the SUT, and the transmission buffer sizes in the SUT. The user can also verify that the SUT service priority assigned to the real TP device(s) is identical to the priority assigned to the emulated devices. Reviews such as these can increase the chances that the test reflects the data communication interface desired by the user.

(4) Review and/or change the contents of SUT data files. Before each benchmark mix execution, a user can review selected portions of all or some data files that are stored on the SUT and that will be used by scenarios. If carefully done, a user can also change portions of selected SUT data files. All changes must not appreciably affect the execution characteristics of the scenarios. The changes can be made immediately before the start of each benchmark mix execution when little time is required to change each data file; e.g., execute interactive commands to replace a few data elements. However, it is mandatory that the changes be made only once and on the first day of the LTD when the changes require large amounts of time to complete; e.g., reload significant portions of a data base. Reviewing and/or changing the contents of SUT data files can help a user determine whether or not all scenarios were successfully completed during a mix execution.

(5) Review and/or change the contents of input data tables. A user can define scenarios that select entries from tables of character strings stored on the RTE and that incorporate the entries in input transmissions sent to the SUT. Before each mix execution, a user can review the contents of one or more input data tables. On the first day of an LTD, a user can also change the contents of all or selected input data tables. It is mandatory that such changes do not appreciably affect the scenario execution characteristics. Reviewing and/or changing the contents of input data tables can reduce the likelihood that a vendor has taken advantage of certain characteristics of the benchmark mix to improperly improve SUT performance.

d. During benchmark mix execution.

(1) Real remote device. A user can require vendors to install and operate one real remote device for each type of device emulated, up to a total of five real remote devices. When the LTD documentation is initially released to vendors, a user can require that a particular

scenario be completed on a real remote device. Using a stopwatch, a user can time certain aspects of the execution of the scenario on a real device. These manual timings alone should not be used to determine whether or not required performance levels were achieved during a mix execution, because in almost all cases too few timings can be taken manually for the results to be a statistically valid indication of the performance levels for the total benchmark mix execution. However, the timings can indicate how thoroughly a user should examine the performance levels recorded in the RTE log files. The use of a real remote device can also help verify that the dialogue properly reflects the scenario.

(2) Benchmark mix execution status information. A user can obtain several types of information about the benchmark mix execution from the RTE system during the mix execution. Every 10 minutes, the RTE will identify automatically any emulated remote device suspected of having incorrectly stopped exchanging messages with the SUT. Upon request, a user can obtain the general status of all emulated remote devices and the status of a specific remote device. (See chapter 5, part 4, paragraph 15 for a complete description of the status information available.) With no advance notice to a vendor, a user can request and obtain status information at any time during a benchmark mix execution. This information can help a user (a) determine if all emulated remote devices were represented correctly during a mix execution and (b) identify those portions of the RTE log file that should be examined in detail after completion of the mix execution.

(3) Data communication line monitor. A user can require the use of one or two data communication line monitors during a benchmark mix execution, and can select the specific data communication link(s) to be monitored immediately before the start of a mix execution. (See chapter 5, part 3, paragraph 12.) A line monitor is a portable device that is independent of both the RTE and SUT and is designed specifically to analyze data communication links and data transmissions. The analysis capabilities of these devices vary by make and model, but example capabilities include (a) displaying in real time all application and control data characters transmitted in both directions over a data communication link; (b) recording all data characters transmitted over a link; (c) timing the durations between the transmission of certain predefined data characters; (d) counting the occurrences of certain data characters in transmissions; and (e) displaying, timing and/or counting certain events after

the completion of a mix execution by replaying in real time all data characters transmitted over a data link. Because a line monitor is independent of the RTE and SUT and is designed not to affect the data link being monitored, this verification technique can accurately determine the character set, line protocol, and messages actually used during a mix execution. The technique can also verify the accuracy of the time-stamps in the RTE. A user should employ a line monitor, however, only when such detailed verification is necessary and when the user has personnel who have detailed experience with these devices. Before the LTD, a user should also practice with the line monitor using a real remote device and example dialogue.

e. After benchmark mix execution.

(1) Review the contents of SUT data files. After each mix execution, a user can review selected portions of all or some SUT data files that were modified by scenarios. This verification technique can help a user determine if all scenarios were successfully completed during a mix execution.

(2) RTE log analyses. Chapter 5, part 4 defines the analyses that a user can require a vendor to perform on the RTE log file(s) created during a benchmark mix execution. A user can require any combination of these analyses for use in verifying a benchmark test. The RTE Log Summary Report provides the most detailed data for verification. After the mix execution, a user can request this report for one or more emulated devices chosen, at that time, either at random or because of some indication of a potential problem; e.g., mix execution status information or line monitor display. (A user should not request this report for all emulated devices because of the enormous volume of the resulting output.) The RTE Log Summary Report can also be used to produce a listing of only the RTE log messages directed to the log by scenarios. Other available analyses include an RTE Log Summary Tape and a listing of all RTE operator console activity. Except for the RTE Log Summary Report, all RTE log analyses shall be defined in detail in the LTD documentation provided vendors.

12. Benchmarking for installation verification.

a. A user occasionally requires a benchmark test that had been conducted during the LTD to be repeated on the initial system configuration installed at the user site. The objective of the test repetition is to verify that the initially installed system has at least the same capacity as the system tested during the LTD, i.e., installation verification. The test repetition is performed by vendor personnel, is conducted before the start of the contractual acceptance test period, and is not a part of the acceptance test procedures.[8] When remote terminal emulation is used during a benchmark test, however, an agency shall not require a vendor to repeat the emulation benchmark test at the agency's site. This restriction is needed primarily because (1) most users do not have sufficient hardware at their sites to configure both the RTE and SUT for the test, and (2) the use of an RTE not colocated with the SUT is usually very costly and time consuming and produces major operational and technical problems; e.g., data communication line errors, delays. (An agency shall contact GSA if, due to extraordinary circumstances, it desires an exception to this restriction.)

b. During the development of a benchmarking strategy, each agency should decide if and how to repeat a benchmark test for installation verification. The choice is important to the total benchmarking strategy and should be made only after careful consideration of the effects of the decision on the benchmarking goal levels desired and the levels of mission and cost risks arising from the decision.

c. When remote terminal emulation is used during a capacity test, and when the agency decides to use benchmarking for installation verification, the agency shall design a separate benchmark test without remote terminal emulation. The following are the most common approaches.

(1) Immediately following the successful completion of the emulation benchmark test during the LTD, the vendor repeats the test without the RTE and the workload demands imposed by it. The agency records the elapsed time required to execute the "new" benchmark mix. During installation verification, the vendor repeats the test, also without the RTE, and the agency again records the required elapsed time. The contract requires that the second elapsed time must not be exceeded by more than X percent the elapsed time recorded at the LTD. The specific percentage reflects the inherent variability of TP systems, the elapsed time of the test, the number of human operator actions in the test, and the chance

---

[8] General Services Administration, "Solicitation Document for ADP Systems," in Standard Solicitation Documents (Washington, DC: GSA/ADTS, continuously updated), pp. E7-E8.

of minor but necesssary differences between the systems; e.g., OS software corrections. The percentage is usually between 5 and 10 percent. This approach is recommended when the original emulation benchmark test contains a sizable volume of local batch workload demands.

(2) Immediately following the successful completion of the emulation benchmark test during the LTD, the vendor conducts a separate test using a local batch mix (and perhaps one or two real TP devices) designed for installation verification. The agency records the elapsed time required to execute the mix. During installation verification, this second test is repeated by the vendor, and the agency again records the elapsed time. The contract requires that the second elapsed time must not exceed the first by more than X percent. This approach is almost identical to the first, except that a separate local batch mix is used, and should be employed only when the emulation benchmark test contains little or no batch workload demands.

d. For installation verification, users occasionally have required that vendors repeat an emulation benchmark test using a vendor-provided driver that executes within the SUT; i.e., an internal driver. This approach is not recommended because (1) all vendors do not have internal drivers, (2) this approach increases vendor time and cost and reduces the probable level of competition, and (3) most users can adequately verify, without the use of an internal driver, that the capacity of an installed system configuration is the same as the capacity of the system tested during the LTD.

PART 2. PREPARATION OF TELEPROCESSING ELEMENTS FOR BENCHMARK TESTS

13. General.

a. In this step of the emulation benchmarking process, a user designs, constructs, tests, and documents the individual TP elements for each capacity test. Five categories of TP elements which typically must be prepared for emulation benchmark tests are (1) TP applications, (2) scenarios, (3) TP device and data communication link configuration, (4) test data used by TP applications and scenarios, and (5) test procedures associated with individual TP elements. A user prepares these elements iteratively, using the benchmarking strategy as the framework for their preparation. This benchmarking step sometimes requires more user time and funds than any other step, except, perhaps, for workload definition and analysis. The TP elements prepared in this step can affect dramatically the achievement of all the benchmarking goal levels. A user, therefore, should ensure that each TP element is prepared in accordance with both the user's benchmarking strategy and the benchmarking goal levels chosen.

b. This part discusses two of the categories of TP elements which are particularly significant to the use of emulation: (1) Scenarios and (2) configurations of TP devices and data communication links. Some of the major factors that a user should consider when preparing these TP elements for emulation benchmark tests are presented. The preparation of these elements depends on a thorough understanding of the remote terminal emulation and benchmarking capabilities described in chapter 5, and the discussions below assume that the reader has studied that chapter. Chapter 5 also contains specific suggestions on when and how to use certain emulation and benchmarking capabilities, along with the definitions of these capabilities.

c. The discussions in this part concentrate on factors that are particularly important to the use of emulation, and complement the guidance in FIPS PUB 42-1, primarily Section III.B (Analysis, Design, Construction, and Documentation of the Benchmark Package, pp. 12-14) and Section IV.B (Construction and Validation of the Benchmark, pp. 18-20).

14. Scenarios. In emulation benchmark tests, scenarios usually are the most important TP elements, primarily because of their enormous impact on both benchmark representativeness

and agency time and costs. The specific actions that a user should take to prepare each scenario depend on the unique circumstances of the acquisition and reflect the user's benchmarking strategy. Users typically should perform five basic tasks during the development of each scenario:

(a) Develop a scenario profile;

(b) Develop and test on some TP system a prototype dialogue for the scenario;

(c) Establish preliminary performance levels for the scenario;

(d) Write an unambiguous, vendor- and SUT-independent description of the workload demands, based on the prototype dialogue; and

(e) Validate the scenario.

a. Develop a scenario profile.

(1) A scenario profile is a list of the specific characteristics that a user desires for a scenario. A user should develop a scenario profile for every scenario to be used in each capacity test, based in part on both the generic types of workload demands chosen for the test and the scenario-workload correspondence established during the development of the benchmarking strategy. The specific types and values of the entries in a scenario profile are determined after analyzing the appropriate degree of similarity between the workload demands in the scenario and the workload demands in the user's projected operational environment. Several different entries can be in a scenario profile, including:

(a) The generic TP application type;

(b) The specific name of an application;

(c) The types, numbers, and sequences of generic user functions;

(d) The types, numbers, and sequences of specific input commands or transactions;

(e) The types and numbers of data files/bases;

(f) The statistical characteristics of think time and typing rate;

(g) The numbers and average sizes of input and output transmissions;

(h) The amount of computer resources to be used by the scenario; e.g., CPU time, disk I/O operations, memory occupancy;

(i) The generic type of TP device used; and

(j) The specific characteristics of the TP device.

(2) The number of different entries in each scenario profile, as well as the specific types and values of the entries, enormously affect benchmark representativeness, the quality of system sizing, the time and cost of the acquisition, the level of competition, and the magnitude of benchmark discrepancies. Increasing the number and types of entries potentially can increase representativeness and the quality of system sizing. In most cases, however, it is extremely time-consuming and costly, if not impossible, for a user to prepare a scenario that matches a large number of entries in a scenario profile. The likelihood of unintentionally biasing a scenario for some vendor or SUT, and thus reducing competition, also increases as the number and type of entries increases. A user, therefore, must prepare each scenario profile with extreme care.

(3) It is recommended that a user employ functional, vendor-independent entries in each scenario profile, and that each scenario profile include at least the generic TP application, the generic TP device, the number and types of user functions, statistical descriptions of think time and typing rate, and the approximate total elapsed time. A user typically should avoid resource-oriented entries, such as CPU time and the number and sizes of transmissions. Moreover, a user should omit any entries if the cost-effectiveness of including that entry is questionable.

b. Develop and test a prototype dialogue.

(1) For each scenario in a capacity test, a user should develop and test on some TP system a prototype dialogue that matches as closely as possible the scenario profile. The characteristics of the prototype dialogue depend on the TP system and TP device used and include all TP operator inputs, actions, pauses, and decisions needed to match the scenario profile. A user also must prepare any

data files and user TP applications needed by the prototype dialogue. The prototype dialogue is only an intermediate step in the scenario development process. The dialogue ultimately used by each vendor almost always differs from the prototype, unless the user provides the TP application. (Some differences often exist even in this case; e.g., logon sequence, prompt character.)

(2) The feasibility and difficulty of developing a prototype dialogue depend on many factors, including the operational status of the TP application (e.g., operational, under development), the complexity of the scenario profile, the availability of a TP system for development, and the approach used during development. The development and testing of a prototype dialogue can be costly and time-consuming, but this helps ensure the practicality and appropriateness of the scenario, assists greatly in documentation, and reduces benchmark discrepancies. It is recommended that a user prepare a prototype dialogue for every scenario in every capacity test. If, for some reason, a user is unable to develop and test a prototype dialogue for some scenario, the user should (a) compare the value of the scenario to the capacity test with the risk of using a scenario that has not been tested in prototype version and (b) omit the scenario from the test in all but the most unusual cases.

(3) Users typically use one or more of three basic approaches to develop each prototype scenario. One approach is for a member of the agency's benchmarking team to compose the dialogue based entirely on the scenario profile. The team member may or may not be familiar with the generic application, or the types and sequence of inputs and actions commonly performed. Another approach is for a team member to work closely with a TP device operator (end user or application designer) during the development of the prototype dialogue. This approach often increases the similiarity between the prototype dialogue and the (projected) operational environment, but also increases the agency's time and cost to prepare the scenario. The third approach is to record actual operational dialogues and to select and/or modify a dialogue that closely matches the scenario profile. It is often very difficult and time consuming, however, to identify an operational dialogue that closely matches a specific scenario profile. This last approach also is not feasible if the TP application is not operational, and may encounter privacy and/or security limitations. The best approach or combination of approaches depends on the unique circumstances of the acquisition, benchmark test, and scenario.

(4) The specific characteristics of the prototype dialogue should reflect the benchmarking goal levels desired and the user's benchmarking strategy. The specific characteristics of a prototype dialogue depend not only on the scenario profile, but also on many features of the TP system, TP application, and TP device used to develop the prototype. A user, therefore, in developing a prototype dialogue should use only TP system, application, and device features that are mandatory features in the RFP, and should avoid features that either are unique to the circumstances of the prototype development or are optional features in the RFP. A user also should minimize the complexity of each prototype dialogue to reduce the benchmarking time and cost for the user and vendor. In addition, a user should ensure that the characteristics of the prototype dialogue are consistent with the emulation capabilities defined in the handbook. The value of a prototype dialogue is reduced when it includes operator actions, TP device features, etc. that cannot be represented during a capacity test by an RTE. The user should study thoroughly chapter 5 before preparing a prototype dialogue, because chapter 5 contains both (a) the definitions of the operator actions, TP device features, etc. that can be represented, and (b) suggestions on when and how to use certain emulation capabilities.

c. <u>Establish preliminary performance levels</u>. A user should establish preliminary values for the performance measures associated with the scenario, based on the performance levels desired during the life of the new system. A user should also employ, if possible, the performance levels obtained while testing the prototype dialogue to help determine whether or not the preliminary performance levels are realistic. For example, the user should measure the total elapsed times required to complete the prototype dialogue during several tests, and should use these values to help determine reasonable scenario turnaround time requirements. The user defines and finalizes these preliminary performance values during the validation of the scenario and validation of the benchmark test.

d. <u>Write a description of workload demands</u>.

(1) A user should draft an unambiguous, vendor- and SUT-independent description of the workload demands in each scenario. Each description:

(a) Is based on the prototype dialogue;

(b) Is in an English narrative format, rather than COBOL, formal grammars, etc.;

(c) Presents the workload demands in totally functional terminology, if possible;

(d) Usually includes only a few of the actual character sequences to be transmitted and/or received by the TP device using the scenario; e.g., transaction codes, file names; and

(e) Includes actual dialogue only when the scenario interacts with a user-provided, interactive application.

(2) By definition, each such description is a scenario, and, when given to vendors, is the definitive statement of the TP workload demands required by the scenario. Appendix B contains an example scenario for a text-editing application.

(3) A user typically should perform the following actions to develop the narrative description for each scenario:

(a) Determine the generic function performed by each input transmission or group of input transmissions in the prototype dialogue;

(b) Develop a narrative description of each generic function;

(c) Eliminate from the narrative any and all vendor and SUT dependencies and bias that are not directly caused by the mandatory requirements in the RFP;

(d) Eliminate ambiguity from the narrative; and

(e) Augment the narrative with any appropriate additional information; e.g., think time, typing rate, user function indicators.

(4) Ambiguity in a scenario can increase the number of benchmark descrepancies and both user and vendor time and costs. A user can reduce ambiguity by choosing with extreme care the words and phrases used in the narrative and by the preparation and use of a thorough technical

glossary. Another important way to eliminate ambiguity is to provide vendors an example dialogue of each scenario. It is recommended that the user provide vendors an example dialogue that is annotated to cross-reference the narrative in the scenario. The user should also identify for vendors the make and model system, TP software utilities, TP application, etc. used to develop the example dialogue. With little or no modifications, the prototype dialogue can be used as the example dialogue. The user provides the example dialogue to vendors only for illustration and guidance, primarily to prevent ambiguities. The user must always understand, and clearly state to the vendors, that, in all cases, the English narrative takes precedence over the example dialogue. In addition to a text-editing scenario, appendix B includes an annotated, example dialogue and implementation instructions for the scenario.

e. Validate the scenario. A user should validate the scenario to ensure that it, the implementation instructions, and the associated preliminary performance levels are realistic, vendor- and SUT-independent and unambiguous. The best way to validate a scenario is for the user to provide to individuals not involved in the preparation of the scenario, the scenario, example dialogue, implementation instructions, and any additional information; e.g., appropriate TP device and link. These individuals should prepare a dialogue implementing the scenario on one or more TP systems different from the system used with the example dialogue. The performance levels achieved while testing these dialogues can indicate whether the preliminary performance levels are realistic for different SUT's. The questions, difficulties, and mistakes arising from this exercise can be used to improve and finalize the implementation instructions and scenario.

15. TP device and data communication link configuration.

a. General.

(1) As part of the benchmarking strategy, the user identified the generic types of TP devices to be included in each capacity test. In this step, the user must determine the configuration of TP devices and data communication link; i.e., the number, types, and detailed characteristics of real and emulated TP devices and data communication links. By increasing the complexity and number of TP devices and links required for a capacity test, a user occasionally can increase benchmark representativeness and uniformity and

improve the quality of sizing. As the complexity of the TP device and link configuration used in a test increases, however, the likelihood of benchmark discrepancies and vendor time and cost also increase. For example, a large number of links increases the hardware required for a test and the chance of a link failure during a test. In addition, the emulation of complicated TP device types and characteristics (e.g., formatted screen capabilities) and large numbers of TP devices increase greatly the size and cost of the required RTE. A user, therefore, should carefully analyze the complexity and level of specificity of the TP device and link configuration required for each benchmark test, and should simplify the configuration whenever possible.

(2) To determine the configuration necessary for each capacity test, a user should list (a) the generic types and specific characteristics, if any, of the generic TP devices chosen during the development of the user's benchmarking strategy, and (b) the numbers and other specific characteristics of each generic TP device and the links projected for the user's operational environment for the contractual life of the acquisition. A user with an existing network of TP devices and links typically can list the TP device and link configuration for each test in greater detail than a user without an existing network. Moreover, when the user requires the vendors to propose a TP device and link configuration, the user should allow each vendor the maximum practical flexibility to determine the configuration for the benchmark test.

b. Simplify the configuration. A user should reduce the complexity of the TP device and link configuration for each test as much as possible by eliminating any specific characteristic that either (1) is not a mandatory feature in the RFP, (2) does not significantly affect benchmark representativeness and the quality of system sizing, or (3) cannot be represented by vendor RTE's and benchmarking facilities. Only TP device and link numbers, types, and characteristics needed to determine the capacity of the SUT should be included. (If necessary, separate functional tests can be conducted to evaluate the SUT's capability to support certain specific characteristics.) A user also should permit vendors the maximum appropriate flexibility to configure TP devices and links to reduce time and costs and increase competition. The amount of vendor flexibility that is appropriate in a test depends primarily on the levels of benchmark representativeness and uniformity desired by the user. Chapter 5 defines the maximum numbers, types, and characteristics of

TP devices and links that vendors can represent during a capacity test. A user should study chapter 5 thoroughly before finalizing the configuration for each test.

c. Validate the configuration. A user should validate the configuration for each test, because it is typically impossible for a user to conduct a trial emulation benchmark test before the actual LTD's. The primary method of validation is to evaluate the combined technical feasibility, interdependencies, and consistency of the (1) TP device and link configuration, (2) the workload demands in the scenarios, and (3) the performance levels required for completing these workload demands. For example, a user should ensure that the user functions and actions in each scenario are consistent with the TP device type and characteristics that will use the scenario; e.g., tab feature, break key, formatted screen capabilities. The required turnaround time for each user function and scenario should also be contrasted with all TP device and link characteristics that affect these timings, e.g., link speed, print speed. A user should calculate the sum of the estimated turnaround times for all scenarios to be completed in a test, and should compare this sum to the product of the permitted elapsed time for the test and the number of remote devices using the scenarios. This comparision can indicate an inconsistency between the scenarios to be completed and the number of remote devices configured to complete the scenarios. In addition, a user should analyze in detail the total data transmission rate specified for each test. The total transmission rate can be calculated by totaling the speeds of all links configured for a single test. (The speed of each full-duplex link should be counted twice.) A user should compare the total transmission rate for each test to the projected operational environment.

d. Document the configuration. Finally, a user should document the numbers, types, and characteristics of all TP devices and links required for each capacity test. A user should state clearly the required (minimum) TP device and link configuration. All vendor options should be explicitly defined. It is recommended that a user document these configuration requirements in table formats. For each generic TP device, a user should prepare a list of the specific characteristics to be represented. For each test, a user should prepare tables that define at least (1) the number of each generic TP device to be installed and emulated, and (2) the number of links of each speed, and whether the links are full- or half-duplex. Chapter 5 includes figures that illustrate possible formats for documenting this information.

PART 3. PREPARATION OF LTD DOCUMENTATION

16. General.

(1) The purpose of this step is to prepare the documentation which describes the LTD requirements to the vendors. This step is one of the most important in the entire benchmarking process. Effective communication is essential to achieving all benchmarking goal levels and is particularly significant in increasing competition, reducing benchmark discrepancies, and decreasing the acquisition time and cost. In this step, the documentation developed in previous steps is combined with additional documentation to form a complete LTD package for release to vendors. The LTD package contains the scenarios and TP device and data communication link configuration discussed in chapter 4, part 2, documentation of several other TP-related areas, documentation of non-TP related areas, and general procedural documentation; e.g., background, objectives, vendor and Government responsibilities, agenda.

(2) This part discusses several areas of the LTD package, and complements FIPS PUB 42-1, Section IV.B (Physical Benchmark Package, pp. 19-20) and Section IV.C (Procedural Documentation and Preparation of the Benchmark for the Vendors, pp. 20-24). The reader should refer to these sections of FIPS PUB 42-1.

(3) The specific areas discussed below are those which are particularly significant in emulation benchmark tests:

(a) Scenarios;

(b) The TP device and data communication link configuration;

(c) TP performance measures;

(d) Vendor emulation reporting requirements;

(e) Benchmark mix execution instructions; and

(f) A technical glossary.

17. Scenarios. For each scenario, documentation in the LTD package should include: (a) The English narrative, (b) implementation instructions and implementation data, (c) a

AD-A082 209 GENERAL SERVICES ADMINISTRATION WASHINGTON DC AUTOMA--ETC F/G 9/2
USE AND SPECIFICATIONS OF REMOTE TERMINAL EMULATION IN ADP SYST--ETC(U)
AUG 79

UNCLASSIFIED FPR-1-4.11 NL

2 OF 3
AD
A082209

sample dialogue, and (d) a scenario overview. An example English narrative with associated implementation instructions and sample dialogue are given in appendix B. The overview should provide information other than implementation instructions or implementation data, and could include a summary, a statement of purpose, etc. A user should provide to vendors in table formats additional implementation data, such as lists of data files required before and after execution, passwords, user-ids, and TP device types on which a scenario may be implemented; e.g., Scenario A must be implemented only on a teleprinter operated asynchronously at 300 bits per second.

18. TP device and data communication link configuration. The user must specify for each benchmark test the TP device and data communication link configuration. The actual configuration for each test was determined in a previous step; in this step this configuration is described to the vendors. The format and complexity of the description will depend on the complexity and level of specificity of the configuration. In general, table formats are recommended; e.g., to specify the minimum number of emulated terminals, a table format such as that shown in figure 4-18 could be used. If a user decides to specify the number of data communication links, and the assignment of TP devices to links, this should be described in table format also.

19. TP performance measures. As part of the development of the benchmarking strategy, the user selected the TP performance measures. The LTD documentation informs the vendors which performance measures will be required, as well as the required performance levels. The user should describe for each test the number of scenarios of each type to be completed in a specified time (throughput), the required turnaround times, and, if used, the response time for application I/O pairs. Again, table formats are recommended. Typically, to describe an increasing TP workload, a user must choose between increasing the number of executions of scenarios required to be completed during each benchmark test and/or decreasing the maximum allowed elapsed time for the execution of each test. Figures 4-19.1 and 4-19.2 give examples of how the user can describe the test requirements when using both techniques.

REMOTE DEVICE TYPE	CHARACTERISTICS	BENCHMARK TEST				
		1	2	3	4	5
1. Interactive Display	Synchronous 2400 bps	70	120	150	175	200
2. Remote Batch	Synchronous 9600 bps	6	15	20	24	28
3. Interactive Teleprinter	Asynchronous 300 bps	140	180	200	215	230

Figure 4-18. Minimum number of emulated TP devices (example format)

SCENARIO	BENCHMARK TEST 1	2	3	4	5
Remote Job Entry (A)	100	140	180	220	260
COBOL Source Edit (B)	14	20	21	23	24
Program Execution (C)	8	8	9	9	10
Program Development (D)	4	9	15	16	17
Text Edit (E)	18	44	105	112	119

Figure 4-19.1. Minimum scenario requirements, by test (example format)

BENCHMARK TEST	MAXIMUM ELAPSED TIME (MINUTES)
1	60
2	55
3	50
4	45
5	40

Figure 4-19.2. Maximum elapsed time, by test (example format)

20. Vendor emulation reporting requirements.

a. The user must describe precisely the emulation-related documentation that vendors must provide during the LTD, as well as when the documentation must be provided; i.e., vendor emulation reporting requirements. Users should only ask vendors for documentation that is necessary for conducting the LTD; unnecessary requests for documentation result in increased time and cost for the user and vendors. The two primary objectives of the required emulation-related documentation are: (1) Summarization and reporting of the

TP performance levels of the SUT, and (2) verification that the RTE portions of the test were conducted in the manner intended by the user.

b. Chapter 5, part 6 describes in detail the formats and contents of three reports that summarize SUT TP performance levels and that all vendors must be able to provide: Scenario Summary Report, Function Summary Report, and Response Time Summary Report. These reports are well-defined and technically consistent for all vendors, and provide sufficient information and flexibility for a user to compare almost all TP systems. It is recommended that users employ only some combination of these reports. If an agency desires additional TP performance reports, however, the agency must prepare any and all RTE log summarization and report generation programs needed to produce the additional reports. All agency-prepared log analysis programs must be in an ANSI standard language, must use the RTE Log Summary Tape (described in chapter 5, part 6) as input, and must be fully described and distributed to vendors in the LTD documentation.

c. Chapter 5, part 6 defines three principal types of verification documentation: (1) The RTE Log Summary Report, (2) the RTE Log Summary Tape, and (3) listings of all RTE operator console activity. It is recommended that a user typically require RTE Log Summary Reports only for a few, selected emulated devices and not for all devices, because of the volume of listings produced for all devices. (The specific devices can be selected by the user immediately after each mix execution). It is also recommended that a user require each vendor to provide a copy of the RTE Log Summary Tape, because this tape can serve as a comprehensive audit trail of the RTE-related aspects of the mix execution. It is also recommended that a user require a listing of all RTE operator console activity.

d. A user also must define clearly all other emulation-related documentation required from each vendor; other documentation could include:

(1) A vendor-signed certification that the vendor's RTE complies with all emulation capabilities required by the user (and defined in this handbook);

(2) Listings of the complete and final dialogue produced from each scenario, annotated to cross-reference the scenario; and

(3) Configuration schematic of the RTE, data communication links, and real TP devices used for each test.

21. Benchmark mix execution instructions. The user must give instructions to the vendors on the execution of the benchmark mix. These instructions must be given in at least the following four areas: (a) Starting and ending conditions, (b) scheduling constraints, (c) scheduling techniques, and (d) verification techniques. The user should describe the intended status, at the beginning and end of each benchmark mix execution, of scenarios, data and program files, TP devices, TP applications, etc., and should identify activities which must be performed before, during, and after the benchmark mix execution. All scenario scheduling constraints should be clearly defined. The user should specify which scenario scheduling technique(s) will be used. Chapter 5, part 4 describes scheduling techniques that a user can require of a vendor's RTE. Chapter 5 specifies that vendors must provide the facility to connect a data communication line monitor to any link configured for a benchmark test. Users may require the use of up to two data communication line monitors during a single test, and should select the specific link(s) to be monitored immediately before the start of a test execution. Users should specify the line monitor requirement in the LTD documentation. The line monitors must be provided by the requesting agency unless a particular vendor wishes to supply them. For verification, vendors must be able to provide, during benchmark mix execution, the benchmark test status information described in chapter 5, part 4. Users should specify the types and maximum number of RTE status requests that the vendor must provide for each test; e.g., at up to ten user-selected times during each mix execution, the user may require upon entry of a single RTE console command: the time of day, the total number of active emulated devices, the number of suspended devices, etc.

22. Technical glossary. A technical glossary with the definitions in this handbook and additional definitions determined by the user must be included in the documentation provided to the vendors. A good glossary helps to clarify all instructions and remove ambiguities. All definitions should be consistent and well-defined. A glossary is particularly important to emulation benchmark tests.

PART 4. AGENCY-VENDOR COMMUNICATION

23. Recommended technical communication.

a. The exchange of technical, benchmarking information between a user and vendor is one of the most important methods of minimizing benchmark discrepancies, maximizing competition, and reducing the time and cost of the acquisition for both the user and vendor. User-vendor communication is more critical in acquisitions involving remote terminal emulation than acquisitions using other benchmarking techniques, because of the increased complexity of emulation benchmark tests. Figure 4-23 describes the minimum recommended technical communication between a user and vendors during an acquisition involving emulation benchmark tests. The actions described are limited to exchanges of RTE-related information and complements the guidance contained in FIPS PUB 42-1, primarily Sections II.C through II.E, pp. 8-9. A user should study FIPS PUB 42-1 and contact his contracting specialists before scheduling meetings with vendors.

b. A user should evaluate carefully the LTD documentation, if any, that the user requires vendors to submit with their proposals. A vendor typically does not complete preparations for each benchmark test earlier than 2 or 3 weeks before that vendor's LTD, because (1) vendor costs can be high to maintain the SUT and RTE hardware and software, as well as the personnel expertise, during the time between the completion of test preparation and the LTD, and (2) there are always scheduling conflicts for vendor benchmarking resources. When a user requires LTD documentation to be submitted with a proposal, vendors may be forced to complete benchmark test preparations before proposal due date, which can be several months before the LTD. This greatly increases the vendors' time and cost and, therefore, decreases the probable level of competition. Alternately, one or more vendors may choose to omit the mandatory LTD documentation from their proposals. A user should not require vendors to include LTD documentation in their proposals unless (1) it is critically important that the user have all the required documentation no later than proposal due date, and (2) the user is prepared to declare non-responsive all proposals that do not contain all the required documentation. (Most users, in fact, can wait until 2 or 3 weeks before an LTD to receive LTD documentation from vendors.)

c. It is recommended that agencies require each vendor to submit LTD documentation, including proof that all tests have been completed successfully on some SUT configuration, at a pre-LTD meeting held with each vendor separately, and after the proposal due date but no earlier than 2 or 3 weeks before that vendor's LTD. (If a user begins LTD's within 2 or 3 weeks of proposal due date, the pre-LTD meetings can begin immediately after receipt of proposals.) Agencies should be aware that some of this documentation may change between the pre-LTD meeting and the LTD, because a vendor has the option during a negotiated procurement to modify the SUT configuration in the vendor's proposal. It is recommended that a user require each vendor to provide complete and final LTD documentation only at a vendor's LTD. It is strongly recommended that users do not require vendors to submit any LTD documentation as a part of their formal proposals. The submission of preliminary LTD documentation on proposal due date should be a vendor option.

WHEN	ACTION
At least 60 days before release of an RFP (This communication is necessary only when a user is granted a waiver by GSA to require emulation capabilities not specified in this handbook.)	User: (a) Provides to requesting vendors detailed instructions specifying all required emulation capabilities not defined in this handbook, as well as the exact manner in which each benchmark test must be conducted; and (b) publishes a notice in the Commerce Business Daily announcing the availability of these instructions.[1]
Within 10 days after release of an RFP	User provides to requesting vendors at least the following LTD Documentation for each emulation benchmark test:[2] (a) Scenarios, each with an annotated, example dialogue; (b) description of the required SUT-RTE data communication interface, if any; e.g., TP device numbers, types, and characteristics, number of links, link protocols and speeds; (c) description of the required real TP devices, if any; (d) source code and documentation for any user-supplied TP applications; (e) data and/or data generation programs and documentation to create data files/bases needed by scenarios and/or TP applications;

[1] User also must submit to GSA certain related material (see FPR Temporary Regulation 49 and Supplement 1 thereto).

[2] See FIPS PUB 42-1 for description of other materials to be provided to vendors.

Figure 4-23. Minimum recommended user-vendor technical communication (Part 1 of 4)

WHEN	ACTION
Within 10 days after release of an RFP (Cont'd)	(f) description of the TP performance measures used and their required values; (g) source code and documentation for RTE log reduction programs, if any, needed to produce any summary report not described in chapter 5, part 6 of this handbook; (h) procedural documentation; e.g., workload scheduling procedures, verification techniques, LTD agenda; and (i) glossary, including the definitions in this document and any additional definitions needed for the acquisition.
After release of the RFP and benchmarking materials, and continuing until 2 or 3 weeks before proposal due date	1. Vendors submit written questions to the user concerning LTD, benchmark tests, remote terminal emulation, etc. 2. User answers vendor questions within five work days, and provides copies of all questions and answers to all participants without identifying (if possible) source of questions.

Figure 4-23. Minimum recommended user-vendor technical communication (Part 2 of 4)

WHEN	ACTION
Not earlier than 1 month after release of RFP and benchmarking materials, but not later than 1 month before proposal due date	User conducts a Preproposal Conference, at which the user: (a) Gives a presentation summarizing the benchmarking strategy and LTD procedures for the acquisition, and (b) answers vendor questions concerning the acquisition, LTD, etc.
On or before proposal due date	Each vendor provides to the user preliminary copies of the following LTD documentation, if available:[3] (a) Description of vendor RTE, including structure, capabilities, etc.; (b) listings of the dialogue produced from each scenario and annotated to cross-reference the scenario; (c) hardware and/or software configuration schematic for each benchmark test, including SUT, RTE, data communication links, real TP devices, etc.; and (d) samples of RTE console listings and requested RTE log reports.

[3] It is impractical to expect and/or require a vendor to have completed preparations for benchmark tests earlier than 2 to 3 weeks before the start of the vendor's LTD. Users, therefore, should not require vendors to submit complete and final copies of LTD documentation on or before proposal due date.

Figure 4-23. Minimum recommended user-vendor technical communication (Part 3 of 4)

WHEN	ACTION
After proposal due date and 2 to 3 weeks before each vendor's LTD	1. User conducts a half-to-full-day, Pre-LTD Meeting with each vendor individually. 2. The vendor: (a) Gives a presentation that at least summarizes the vendor's RTE; describes the configurations of the RTE, SUT, links, and real TP devices used in each benchmark test; outlines the day-to-day progress planned during the LTD; (b) provides complete, updated copies of all the LTD documentation requested by the user;[4] (c) provides additional documentation to show that all benchmark tests have been conducted; and (d) answers all the user's technical questions about the LTD, RTE, and SUT.

[4]The user should expect some changes in LTD documentation until each test is conducted during the LTD.

Figure 4-23. Minimum recommended user-vendor technical communication (Part 4 of 4)

TABLE OF CONTENTS

CHAPTER 5. REMOTE TERMINAL EMULATION SPECIFICATIONS

TABLE OF CONTENTS CONTINUED

PART 6. RTE LOG ANALYSES

TABLE OF CONTENTS CONTINUED

CHAPTER 5. REMOTE TERMINAL EMULATION SPECIFICATIONS

1. Scope.

a. This chapter specifies the remote terminal emulation capabilities that Government agencies are permitted to require vendors to provide for benchmark tests conducted at vendors' facilities during ADP system acquisitions. The specifications are divided into six parts:

(1) Teleprocessing Device Representations;

(2) Terminal Operator Representations;

(3) Data Communication Link Representations;

(4) RTE Driver Characteristics;

(5) RTE Monitor Characteristics; and

(6) RTE Log Analyses.

b. An agency is permitted to require a vendor to provide only those remote terminal emulation capabilities needed to validate the capacity of the SUT hardware and software components actually bid by the vendor. For a particular acquisition, an agency may require all vendors to provide the emulation capabilities needed to validate the capacity of the mandatory support items. For desirable (optional) support items, an agency shall require only those vendors who bid the desirable items to provide the emulation capabilities used to validate the capacity of the desirable items. Regardless of the mandatory and desirable SUT support items, however, an agency shall not require emulation capabilities that are not explicitly defined in this chapter. An agency shall also not require vendors to provide any emulation capabilities for benchmark tests at the agency's site; e.g., pilot tests, installation verification. When the benchmark instructions are released to industry, moreover, an agency shall define clearly the emulation capabilities that vendors must provide. An agency is permitted to declare a vendor nonresponsive and to disqualify the vendor from the acquisition, if the vendor does not provide the subset of the emulation capabilities explicitly specified in this chapter that the agency determines are needed to validate the SUT. This chapter defines the maximum set of remote terminal emulation capabilities that an agency may require of vendors.

c. To qualify to bid on most Government ADP system acquisitions, a vendor must possess at least the remote terminal emulation capabilities specified in this chapter. A vendor may have greater emulation capabilities than specified herein. If a vendor never bids a certain SUT support item, however, the vendor may choose (with impunity) not to have the emulation capability necessary to validate that support item. In all cases, a vendor retains the right to request from the procuring agency a waiver of any benchmark test and/or remote terminal emulation requirement.

2. Audience. The objective of this chapter is to define clearly the remote terminal emulation capabilities that (1) Government agencies are permitted to require vendors to provide for benchmark tests during ADP system acquisitions and (2) should be common to all computer system vendors participating in Federal TP system acquisitions. The intended audience of this chapter, therefore, includes both agency and vendor personnel. The manager of an agency's benchmark development project and each agency benchmark analyst must understand thoroughly these capabilities before designing, implementing, and conducting emulation benchmark tests, because an agency is forbidden to require emulation capabilities that are not explicitly defined in this chapter. Vendor benchmark management and technical personnel also must understand the capabilities specified in this chapter, because a vendor must provide the management and technical resources needed to plan for, implement, maintain, and employ as many of these capabilities as the vendor chooses to provide.

3. Admonition.

a. An agency should require in an emulation benchmark test as few emulation capabilities as needed to validate SUT capacity. The required capabilities should also be kept as simple as possible. Increasing the number and complexity of emulation capabilities in a test results in increased agency and vendor time and costs and a higher probability of benchmark discrepancies. Because some vendors may choose not to provide every emulation capability specified in this handbook, an agency also reduces the probable level of competition when the agency increases the emulation capabilities required for a test.

b. Vendor compliance with the remote terminal emulation capabilities specified in this chapter will not be validated or certified by a central Government group. At the time of an LTD, therefore, an individual agency should (1) require vendors to sign a statement certifying that the emulation capabilities employed comply with the portions of these specifications required by the agency; and (2) use any and all reasonable methods to verify that the benchmark test, including the RTE, is conducted according to the agency requirements. When an agency and a vendor disagree on the correct interpretation of these specifications, however, either party may request GSA to arbitrate the disagreement in a timely manner. An agency must also obtain a waiver from GSA if, because of extraordinary circumstances, the agency desires to require vendors to provide emulation capabilities that are not explicitly defined herein. (See FPR Temporary Regulation 49 and Supplement 1 thereto.)

PART 1. TELEPROCESSING DEVICE REPRESENTATIONS

4. General.

a. This handbook defines two broad categories of teleprocessing devices that a Government agency may require vendors to represent during a benchmark test. One category, referred to as remote devices, is composed of those TP devices where user work units originate. Example remote devices include interactive teleprinters, remote batch terminals, and remote host systems. The other category, referred to as intermediate devices, is composed of those TP devices used to connect remote devices to a host computer system. When used, intermediate devices are configured between remote devices and host systems. Example intermediate devices include terminal cluster controllers and concentrators.

b. The numbers, types, and specific characteristics of remote and intermediate devices that an agency is permitted to require vendors to represent (physically install or emulate) for a benchmark test depend, in part, upon whether or not vendors bid the devices as part of their proposals. This part, therefore, is divided into subsections describing the emulation capabilities that apply to (1) specific make and model TP devices, (2) generic types of TP devices, and (3) both specific make and model and generic types of devices. At the time of benchmark mix execution, however, an agency should require vendors to certify the exact numbers, makes, and models of TP devices both physically installed and emulated.

5. Representations of specific make and model TP devices. When a vendor bids TP devices as part of its proposal, an agency may require the vendor, during a benchmark test, to represent the total number and exact make(s), model(s), and operational characteristics of the bid devices. An agency may require a vendor to install and operate up to two real TP devices of each make and model bid, provided that the total number of real remote devices is not more than five. An agency also may require a vendor to emulate, instead of install, any remote device that was bid when more than five of such devices must be represented during a test. (A vendor must emulate the same remote device characteristics when the remote device connects to the SUT either directly or through an intermediate device.) Vendors, however, may elect to install, instead of emulate, any make and model

remote device when less than six of such devices must be represented. Vendors may also elect to install the intermediate devices bid, regardless of the total number to be represented.

6. Representations of generic TP devices.

a. Generic remote devices. Whether or not a vendor bids TP devices as part of its proposal, an agency may require the vendor, during a benchmark test, to represent any number and combination of the six types of generic remote devices defined in figure 5-6.1 (located at the end of this part). An agency may also require the representation of any combination of the options, peripherals, character sets, etc. listed for each generic remote device type. Agencies shall state clearly which agency-defined types and options of generic remote devices are to be represented, and which choices are vendor options. For example, an agency might allow each vendor to choose the synchronous protocol used by interactive synchronous displays. An agency may require a vendor to install and operate up to two real remote devices that match each generic type, speed, character set, etc. represented during a test, provided that the total number of real remote devices is not more than five. An agency may also require a vendor to emulate, instead of physically install, any generic remote device when more than five of such devices must be represented. (A vendor must emulate the same generic remote device characteristics when the remote device connects to the SUT either directly or through an intermediate device.)

b. Generic intermediate devices. An agency may require a vendor, during a benchmark test, to represent any number and combination of the three types of generic intermediate devices and device characteristics listed in figure 5-6.2 (located at the end of this part). Agencies shall state clearly which agency-defined types and options of generic intermediate devices are to be represented, and which choices are vendor options. Vendors always may choose to install and/or emulate generic intermediate devices represented during a test.

c. Future standards. In addition to the generic TP device types and options explicitly specified in figures 5-6.1 and 5-6.2, agencies may require vendors to emulate any TP device character set and data communication protocol one year after its formal adoption as a standard by either the

American National Standards Institute (ANSI) or the Federal Information Processing Standards (FIPS) program.

7. Representations of common TP device characteristics.

a. Typeahead. An agency shall not require vendors to emulate typeahead on interactive remote devices. For certain applications and computer systems, however, the use of typeahead can significantly affect turnaround time and throughput and may reduce the comparability of benchmark test results. An agency should analyze the impact and feasibility of typeahead in its expected operational environment before deciding whether or not to allow vendors the option of emulating typeahead.

b. Data encryption/decryption. An agency shall not require a vendor, during a benchmark test, to represent remote devices equipped with data encryption/decryption capabilities when encryption/decryption is implemented in hardware in the vendor's proposed system. When a vendor bids a system that employs software encryption/decryption, however, (1) an agency may require the vendor to represent remote devices equipped with this capability, and (2) the vendor may elect either to install encryption/decryption hardware on the RTE system and/or to emulate the encryption/decryption components.

c. Print time. Agencies may require vendors to emulate the print time of, and any associated delays caused by, (1) interactive synchronous teleprinters, (2) character printers attached to interactive synchronous displays, and (3) remote batch terminal (RBT) printers. For non-RBT printers, agencies shall either specify or allow vendors to choose the print rates in characters per second; vendors will calculate print times by dividing the number of printable characters in each transmission (asynchronous line or synchronous block) by the specified print rate. Any operator delay or terminal "lockout" caused by the print time must be emulated; e.g., postponing the start of think time for the next operator input until the print time has elapsed. Vendors will ensure that the SUT transmits a typical number, if any, of null or "pad" characters to allow for carriage return, line feed, page eject, etc. For RBT printers, agencies may specify (or allow vendors to choose) the print rate in lines per minute. The print rate will be established by assuming all print lines are of an agency-specified average length. Vendor RTE's will calculate print time by dividing the number of print lines (regardless of actual length) in each output transmission block by the specified print rate. All flow control protocol transmissions from all printers must be emulated, and no remote device may accept print blocks from the SUT faster than the emulated printer can "print" them.

d. Card input time. Agencies may require vendors to emulate card input time for RBT card readers. Agencies shall specify (or allow vendors to choose) the card input time in cards per minute; vendors will calculate card input time by dividing the number of card images in each transmission block by the specified input rate. The emulated RBT may not transmit card images faster than the emulated card reader can "read" them, and any input delays must be imposed between each transmission block.

e. General.

(1) Each vendor will restrict the size of transmission blocks to the size(s) that the vendor's proposed system commonly use(s) in operational environments similar to that of the procuring agency. For example, very large blocks usually would not be used over switched, 4800 bits per second (bps) circuits because of the expected error rate. Agencies should require vendors to document the transmission block size(s) used in each benchmark test execution, and should verify the size(s) by such means as spot checks of the RTE log reports, data communication line monitor displays, etc.

(2) Agencies should require vendors, during each benchmark test, to use device and communication control software in the SUT which is identical to that proposed for the operational system; e.g., front-end processor executives, line protocol handlers, device handlers, access methods. Further, vendors should configure software options that are often used in operational environments similar to that of the procuring agency; e.g., data compression, poll lists and rates, number and sizes of buffers. Exceptions to this requirement may be necessary because vendors occasionally can be required to emulate only agency-specified, generic device types and characteristics, not the specific makes and models to be used in the operational system. Agencies and vendors should make every effort to minimize these exceptions. Agencies should require vendors to document the specific software used in each benchmark test, describe all differences from the proposed operational system, and certify that the differences have not improved SUT performance.

(3) Agencies shall permit vendors to modify standard SUT software, so that a special, nonprintable character is sent to an emulated asynchronous device at the completion of the SUT output in response to each device input.

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
1. Interactive Asynchronous Teleprinter	Upper and lower case ASCII character set
	"TTY" asynchronous line protocol
	Two-way alternate transmission (e.g., no Echoplex)
	Operation over dedicated, point-to-point communication link only (no dial-up, multipoint, or polled capabilities)
	Speeds in bits per second (characters per second): 110(10), 300(30), 1200 (120), 2400(240), 4800(480), 7200 (720), 9600(960)
	Up to 132-character print line
	Buffered or unbuffered
	Page eject capability

Figure 5-6.1. Types of generic remote devices to be emulated (Part 1 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
1. Interactive Asynchronous Teleprinter (Cont'd)	Horizontal tab feature
	Break key feature
	No peripheral attachments (e.g., no cassette)
2. Interactive Asynchronous Display	Upper and lower case ASCII character set
	"TTY" asynchronous line protocol
	Two-way alternate transmission (e.g., no Echoplex)
	Operation over dedicated, point-to-point communication link only (no dial-up, multi-point, or polled capabilities)

Figure 5-6.1. Types of generic remote devices to be emulated (Part 2 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
2. Interactive Asynchronous Display (Cont'd)	Speeds in bits per second (characters per second): 110(10), 300(30), 1200 (120), 2400(240), 4800(480), 7200 (720), 9600(960)
	Eighty characters per display line
	Up to 24 display lines
	No local paging capability
	Clear display feature
	Horizontal tab feature
	No forms-mode operation (e.g., no cursor positioning control, no fixed/variable fields)
	Break key feature
3. Interactive Synchronous Teleprinter	Upper and lower case ASCII character set
	ANSI X3.28-1976 or ANSI X3.66-1979 (ADCCP)* synchronous line protocol

Figure 5-6.1. Types of generic remote devices to be emulated (Part 3 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
3. Interactive Synchronous Teleprinter (Cont'd)	Two-way alternate transmission
	Operation over dedicated point-to-point or dedicated multipoint communication link
	Single device or cluster configurations
	Speeds in bits per second: 1200, 2400, 4800, 7200, 9600
	Up to 132-character print line
	Transmission buffer size less than 300 characters
	Page eject capability
	Horizontal tab capability
	No peripheral attachments (e.g., no cassette)

Figure 5-6.1. Types of generic remote devices to be emulated (Part 4 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
4. Interactive Synchronous Display	Upper and lower case ASCII character set
	ANSI X3.28-1976 or ANSI X3.66-1979 (ADCCP)* synchronous line protocol
	Two-way alternate or two-way simultaneous transmission
	Operation over dedicated point-to-point or dedicated multipoint communication link
	Single device or cluster configuration
	Speeds in bits per second: 1200, 2400, 4800, 7200, 9600
	Eighty characters per display line
	Up to 24 display lines
	Transmission buffer size less than 2200 characters

Figure 5-6.1. Types of generic remote devices to be emulated (Part 5 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
4. Interactive Synchronous Display (Cont'd)	No local paging capability
	Clear display feature
	No program function keys
	Horizontal tab feature
	Forms mode operation, with fixed and variable fields and cursor positioning controls
	Character printer attachment for printed copy of the display; printer options: Speeds identical to display speeds Eighty characters per print line Keyboard locked while printing

Figure 5-6.1. Types of generic remote devices to be emulated (Part 6 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
5. Remote Batch Terminal	Upper and lower case ASCII character set
	ANSI X3.28-1976 or ANSI X3.66-1979 (ADCCP)* synchronous line protocol
	Two-way alternate or two-way simultaneous transmission
	Capability to transmit binary data (e.g., transparency)
	Operation over dedicated point-to-point communication link
	Multirecord transmission option
	Speeds in bit per second: 2400, 4800, 7200, 9600, 19200, 40800, 50000, 56000
	Transmission buffer size less 1000 characters
	Maximum peripherals are one line printer and one card reader

Figure 5-6.1. Types of generic remote devices to be emulated (Part 7 of 8)

GENERIC REMOTE DEVICE TYPE	CHARACTERISTICS/OPTIONS
5. Remote Batch Terminal (Cont'd)	Capability for line printer and card reader to operate simultaneously (e.g., multistream, multistring, multileaved)
6. Remote Host Systems	Upper and lower case ASCII character set
	ANSI X3.28-1976 or ANSI X3.66-1979 (ADCCP)* synchronous line protocol
	Support for two-way file and batch job transfers between SUT and emulated Remote Host System
	Two-way alternate or two-way simultaneous transmission
	Operation over dedicated point-to-point communication link
	Speeds in bits per second: 2400, 4800, 7200, 9600, 19200, 40800, 50000, 56000

*Emulation of ANSI X3.66-1979 (ADCCP) may not be required until February 1, 1980.

Figure 5-6.1. Types of generic remote devices to be emulated (Part 8 of 8)

GENERIC INTERMEDIATE DEVICE TYPE	CHARACTERISTICS/OPTIONS
1. Cluster Controller	ANSI X3.28-1976 or ANSI X3.66-1979 (ADCCP)[1] synchronous line protocol
	Two-way alternate or two-way simultaneous transmission
	Supports up to 10 interactive synchronous teleprinters and/or interactive synchronous displays
	Operates over dedicated point-to-point or dedicated multipoint communication link
	Speeds in bits per second: 1200, 2400, 4800, 7200, 9600
	General and specific poll features
2. Concentrator	Supports all of the generic remote device types listed in figure 5-6.1
	ANSI X3.28-1976 or ANSI X3.66-1979 (ADCCP)[1] synchronous line protocol

Figure 5-6.2. Types of generic intermediate devices to be represented (Part 1 of 3)

GENERIC INTERMEDIATE DEVICE TYPE	CHARACTERISTICS/OPTIONS
2. Concentrator (Cont'd)	Two-way alternate or two-way simultaneous transmission
	Operation over dedicated point-to-point communication link
	Speeds in bits per second: 4800, 7200, 9600, 19200, 40800, 50000, 56000
3. Packet Network Interface Device	CCITT X.25 protocol[2]
	Support for all generic remote device types listed in figure 5-6.1
	Two-way alternate or two-way simultaneous transmission

Figure 5-6.2. Types of generic intermediate devices to be represented (Part 2 of 3)

GENERIC INTERMEDIATE DEVICE TYPE	CHARACTERISTICS/OPTIONS
3. Packet Network Interface Device (Cont'd)	Operation over dedicted point-to-point communication link Speeds in bits per second: 4800, 7200, 9600, 19200, 40800, 50000, 56000

[1]Emulation of ANSI X3.66-1979 (ADCCP) may not be required until February 1, 1980.

[2]Emulation of CCITT X.25 may not be required until one year after final adoption by either ANSI or FIPS.

Figure 5-6.2. Types of generic intermediate devices to be represented (Part 3 of 3)

PART 2. TERMINAL OPERATOR REPRESENTATIONS

8. General. Agencies may require vendors, for interactive scenarios only, to emulate terminal operator think times and/or type times. For interactive asynchronous devices, figure 2-12.1 illustrates the relationship of think time and type time to other significant application I/O pair events. Figure 2-12.2 illustrates this relationship for interactive synchronous devices. (These figures are located in chapter 2.)

9. Think time

a. Agencies may define think time value(s) once for each interactive scenario and/or for any specific operator input in an interactive scenario. Vendors will implement think time as a single block delay that occurs immediately after the receipt of the SUT response to the previous emulated operator input. Vendors will use the scenario-level value for all operator inputs in that scenario, except for each specific input that has another value defined. For each scenario, agencies may define think time either as a constant value or by specifying a statistical distribution. For specific operator inputs, agencies may define think time only as a constant value. All think time values will be specified to a precision of one-half second and implemented to a minimum accuracy of one-half second.

b. Agencies may select one of three statistical distributions of think time values for each interactive scenario: truncated negative exponential, truncated gaussian, or uniform. The range of each distribution will be from an agency-specified minimum value to an agency-specified maximum value not greater than 300 seconds. When using the truncated negative exponential distribution, agencies will specify the average, minimum, and maximum values for the distribution. Agencies will specify the average, standard deviation, minimum, and maximum values for the truncated gaussian distribution, and the minimum and maximum values for the uniform distribution.

c. The random number seed used to produce the think time statistical distributions must be accessible to the Government. Agencies may specify the seed value immediately prior to each benchmark test execution. (Because a single random number generator will be used to produce all statistical distributions in each RTE, modifying the seed value in an RTE will affect not only think times but also all other values and actions controlled by statistical distributions in that RTE.)

10. Type time.

a. Agencies may define a constant typing rate in characters per second (cps) for each interactive scenario. It is mandatory that the precision of the rate (if defined) be in tenths of cps; e.g., x.x cps.

b. For interactive synchronous devices, vendors will implement type time as a single delay that occurs after the expiration of any think time delay and before the start of the transmission of the next emulated operator input. Vendors will use the following formulae to calculate the total type time for every operator input from an interactive synchronous device:

If the typing rate is not zero, then

$$\text{Type Time (In Seconds)} = \frac{\text{Number characters in input}}{\text{Typing rate}}$$

If the typing rate is zero or not defined, then

Type Time = 0 seconds

Vendors will calculate and implement type time to a minimum precision of one-tenth second.

c. For interactive asynchronous devices, vendors will implement type time according to the specifications defined above for interactive synchronous devices; i.e., a single, block delay calculated according to the stated formulae and precision requirement. Vendors may also choose to implement type time for interactive asynchronous devices as a series of separate delays, where a delay occurs after the transmission of each character of an emulated operator input; i.e., by intercharacter delays. The value of each intercharacter delay will be the inverse of the agency-specified type rate, calculated and implemented to a minimum precision of one-hundredth of a second. For unbuffered, interactive asynchronous devices, vendors have the option to emulate type time by either a single, block delay or a series of intercharacter delays. For buffered, interactive asynchronous devices, agencies either may require vendors to emulate type time as a single, block delay or may permit vendors to choose the method. Agencies shall not require a vendor to emulate type time by a series of intercharacter delays.

PART 3. DATA COMMUNICATION LINK REPRESENTATIONS

11. Types and numbers of links.

a. Figure 5-11 defines the types and maximum numbers of real data communication links that agencies may require vendors to install between all RTE's and the SUT, and to operate simultaneously for *a single benchmark test execution.* Agencies are not permitted to require any vendor to install over 256 links, regardless of the number of types of TP devices represented and of the number of RTE's used in a benchmark test. The total links needed to support all real and emulated TP devices during any test must not exceed a combination of the types and maximum numbers in figure 5-11. In a single benchmark test, for example, an agency may require vendors to operate simultaneously: (1) 160 half-duplex links, each attached to an emulated, 300 bits per second (bps) interactive asynchronous teleprinter; (2) 52 half-duplex links, each attached to an emulated synchronous remote batch terminal operating at 4800 bps; (3) one full-duplex link attached to an emulated remote host system operating at 19200 bps. The vendor will be free to use any standard electrical link interface that the vendor will certify does not affect performance; e.g., EIA RS-232-C, EIA RS-449.

MAXIMUM NUMBER OF HALF-DUPLEX LINKS[1]	CLASS(ES) OF LINE PROTOCOL	SPEED RANGE (BITS PER SECOND)
150	Asynchronous	110-9600
50	Asynchronous or Synchronous[2]	110-9600
50	Synchronous[2]	1200-9600
6	Synchronous[2]	19200-56000

[1]The maximum number of full-duplex links is half of the maximum number of half-duplex links. Each full-duplex link counts as two half-duplex links.

[2]Either character-oriented or bit-oriented synchronous line protocols

Figure 5-11. Types and maximum numbers of communication links

b. Agencies are cautioned to analyze in detail the total benchmark transmission rate specified for each single test, in addition to the total numbers of links and TP devices. The total benchmark transmission rate can be calculated by totalling the speeds of all links configured for a single test; the speed of each full-duplex link should be counted twice. When designing a benchmark test using an RTE, an agency should evaluate the representativeness of the total benchmark transmission rate.

12. Data communication line monitor.

. Vendors must provide the facility to connect a data communication line monitor to any link configured for a benchmark test. Agencies may require the use of up to two data communication line monitors during a single test, and will select the specific link(s) to be monitored immediately before the start of a mix execution. Agencies shall specify, at the time of the release of the LTD documentation to industry, the required number of monitors, if any, and the functional capabilities of the monitor. It is mandatory that the line monitors be provided by the requesting agency unless a particular vendor wishes to supply them. A vendor that notifies the agency, by proposal due date, of a desire to provide the monitors shall be permitted to do so. In all cases, the vendor will attach the monitors, before the start of a test, to the links selected by the agency. Monitors shall not be switched from one link to another during a mix execution.

b. When provided by the Government, the line monitor(s) must have one or more of the following electrical link interfaces for operation at the indicated link speeds: (1) EIA RS-232-C, for all speeds up to and including 19200 bps; (2) EIA RS-303, for speeds between 19200 bps and 50000 bps, inclusive; and (3) EIA RS-449, for speeds equal to or greater than 19,200 bps. Agencies shall specify the electrical link interface(s) of the Government-provided line monitor(s) at the time benchmark instructions are released to industry. Vendors will supply any and all adaptors needed to attach the Government-supplied line monitor(s) to any communication link.

PART 4. RTE DRIVER CHARACTERISTICS

13. Generation of application input messages.

a. General. Agencies may define scenarios that require vendor RTE's to transmit and/or receive any valid application data sequence that a specified type of remote device could transmit and/or receive. Possible data sequences include, but are not limited to, interactive LOGON procedures, interactive line delete, backspace, and break control characters, interactive requests for the current time of day, binary remote batch input jobs, etc.

b. Input data tables. Agencies may define scenarios that use tables of character strings to create emulated application input messages; e.g., query values, file names. Each character string may be up to 160 characters in length. Multiple scenarios concurrently may use a single input data table. Agencies may require that the entries in a given table be sent to the SUT by the RTE in exactly the sequence specified; i.e., the first scenario that actually accesses the table must use the first table entry, the next scenario must use the second entry, etc. Agencies also may require the entries in a given table to be sent to the SUT in a uniformly random order; i.e., all entries must have an equal probability of being used by the next scenario to access that table, and a single table entry may be used more than once during a test. Agencies must be able to access and modify the random number seed used to create the uniform distribution. Agencies may specify the seed value immediately prior to the start of a benchmark test execution.

c. Date and time of day. Scenarios may be defined that access the RTE system clock and use the current date and/or time of day values within emulated application input messages. The format of the current date value transmitted from the RTE to the SUT will be YYMMDD, where YY is the units and tens identification of the year, MM is the month (01-12), and DD is the day (01-31). (See FIPS PUB 4.) The format of the current time of day value will be HH:MM:SS, where HH is the hour (00-23), MM is the minute (00-59), and SS is the second (00-59). (See ANSI X3.43-1977.) Immediately prior to the start of each benchmark mix execution, agencies may reset the RTE system clock value to any year, month, and day, and to any time of day between 1 am and 9 pm.

d. Data comparison and storage. Agencies may define scenarios that require vendor RTE's, for a specific emulated application input, to (1) compare the resulting SUT output to a predefined data string or numeric value and (2) terminate the current scenario or jump to another portion of that scenario if the two strings are either identical or not identical, or if two numeric values are either equal or not equal. The maximum length of these data strings is 40 characters. To use this data comparison feature, however, agencies must ensure that each occurrence of a SUT output string to be compared always begins in the same character-position of the application data portion of the output message. Agencies also may define scenarios that require vendor RTE's to store up to 40 characters of a specific SUT output and to use those characters within the next device input. (Agencies are strongly urged to avoid using both the data comparison and data storage features, whenever possible. Both features greatly increase the complexity and expense of benchmark tests, can affect the repeatability of the tests, and, therefore, increase the chance of problems and delays during an acquisition. Almost all TP workloads can be represented adequately without either of these features.)

e. RTE log messages. Agencies may define scenarios that send a message of up to 40 characters to the RTE log file. Both the time of day and the unique identifier of the device using the scenario also will be included when such a message is added to the RTE log file. (Chapter 5, part 5 contains descriptions of the RTE log file contents.)

14. Execution of benchmark test procedures.

a. Scenario suspend and restart. Agencies may define benchmark test procedures that require suspending and restarting scenarios. Each scenario must be able to suspend itself between any two user functions, either every time the scenario is used or only the first time it is used on a particular emulated device. The emulated device using a suspended scenario will continue to respond to SUT control messages as an active device would respond; i.e., control I/O pairs will continue. By using commands from the RTE operator console, vendors must be able to (1) suspend all active scenarios, (2) restart all suspended scenarios, (3) suspend all active scenarios used by the device(s) on a single, selected communication link, and (4) restart all suspended scenarios used by the device(s) on a single, selected link. When a scenario is suspended from the RTE operator console, the next application I/O pair will not begin, but any active application I/O pair will be completed. The time of day that one of these commands is executed will be printed or displayed on the RTE console immediately after the entry of the command. The time value will be based on the RTE system clock and should be identical to the time recorded in the RTE log file.

b. Interscenario delay. Agencies may specify fixed or random interscenario delays (the elapsed times between the end of one scenario and the start of the next scenario on the same device) or may allow the vendors to choose the interscenario delays. (A scenario begins at the start of the first application I/O pair of that scenario and ends at the end of its last application I/O pair.) All interscenario delays will be specified to a precision of one-half second and implemented to a minimum accuracy of one-half second. For random delays, agencies will select either a truncated negative exponential, a truncated gaussian, or a uniform distribution. The range of each distribution will be between agency-specified minimum and maximum values, but the maximum may not be greater than 300 seconds. To use a truncated negative exponential distribution, agencies will specify the average, minimum, and maximum values for the distribution. Agencies will specify the average, standard deviation, minimum, and maximum values for the truncated gaussian distribution, and the minimum and maximum values for the uniform distribution. The random number seed used to produce the statistical distributions must be accessible to the Government. Agencies may specify the seed value immediately prior to each benchmark test execution.

c. Scenario scheduling.

(1) Agencies shall specify benchmark test procedures that require vendor RTE's to schedule scenarios (a) in a fixed, agency-specified order from agency-specified emulated remote devices; (b) in a fixed, vendor-chosen sequence from vendor-chosen emulated remote devices of agency-specified types, speeds, protocols, etc.; (c) using the initiation control scheduling technique (defined below); or (d) using an agency-specified combination of these alternatives. In addition, agencies may specify the exact number of scenarios to be scheduled during the mix execution and/or to be used by each device; e.g., Device X will use Scenario Y twice, Scenario Z once, and then stop; exactly 120 scenarios will be scheduled during the mix execution.

(2) When the LTD documentation is released to industry, an agency shall clearly specify which scenario scheduling procedures are to be used in each benchmark test. For each (up to a maximum of 20) agency-defined group of

scenarios, if any, an agency shall specify the scenario scheduling procedure to be used. Each scenario can be in only one scenario group and each emulated device can use the scenarios in only one scenario group.

(3) Initiation control uses the total history of all scenarios started during a benchmark mix execution to keep the cumulative percentage of each scenario scheduled as close as possible to an agency-specified percentage; e.g., 23 percent Scenario X. To use initiation control, an agency shall define at least one scenario group. For each scenario group that uses initiation control, an agency shall also define (a) the number, type, speed, protocol, etc. of the remote device that can use the scenarios in that group, and (b) the desired cumulative percentage of each scenario in that group to be completed during the benchmark test. For each scenario group, agencies may also define the maximum number of all scenarios in that group to be completed; e.g., a total of 40 scenarios in Scenario Group I will be completed.

(4) The following steps define initiation control:

(a) Schedule the first scenario to be initiated for each emulated remote device assigned to scenario group Gk (The initial scenario assignments can be either specified by the agency or left as a vendor option, but should approximate the agency-defined percentages.);

(b) For each scenario Si assigned to group Gk, set a counter NSi to the total number of times that the scenario has been scheduled as the first scenario for any emulated device;

(c) Set a counter TSk to the total number of times all scenarios assigned to group Gk have been scheduled as the first scenario for any emulated device;

(d) Begin the benchmark mix execution, satisfying all other agency-defined test procedures; e.g., suspend and restart;

(e) When any active emulated remote device Dj (assigned to scenario group Gk) completes a scenario, compute Fik for every scenario Si assigned to group Gk:

$$Fik = \frac{NSi}{TSk*PSi}$$

where

NSi = the total number of times that scenario Si has been scheduled on all devices

TSk = the total number of times that all scenarios in group Gk have been scheduled on all devices

PSi = desired percentage of scenario Si

(f) If the total number of all scenarios scheduled during the mix execution equals an agency specified limit, then suspend device Dj and go to step (e);

(g) If the total number scheduled of all scenarios in group Gk equals an agency-specified limit, then suspend device Dj and go to step (e);

(h) If the total number of all scenarios scheduled on Dj equals an agency-specified limit, then suspend device Dj and go to step (e);

(i) Determine the minimum Fik for Dj;

(j) Schedule scenario Si to device Dj, where the value of "i" is the one that corresponds to the minimum value of Fik;

(k) Increment by one the values of NSi and TSk;

(l) Postpone the initiation of scenario Si until any defined interscenario delay has elasped;

(m) Initiate scenario Si on device Dj; and

(n) Gc to step (e).

15. Provision of benchmark mix execution status information.

a. General. Using RTE's, vendors must be able to provide, during benchmark mix execution, the status information described below. Vendors have the option of printing or displaying the information on the RTE console or some similar device. If displayed, the data must remain on the display screen at least 30 seconds. In addition, all status information either must be written to the RTE log file or must be provided to an agency in printed form after the completion of a benchmark test. (Chapter 5, part 5 describes the RTE log file contents in detail.)

b. Remote devices suspected of having stopped. Every 10 minutes certain status information must be provided without a specific RTE console request if any active (nonsuspended) emulated remote device has neither sent nor received an application data transmission for an agency-specified length of time. The following information must be provided: the current time of day, the unique identifier of each remote device suspected of having stopped, the communication link to which each such remote device is assigned, the scenario that each remote device is using, the time of the last application data input for each device, and the time of the last SUT output to each device. For each benchmark test, an agency may specify the length of time (in integer minutes between 1 and 10) during which a device must not have sent or received application data before the device will be suspected of having stopped.

c. Remote device status.

(1) General device status. Upon entry of a single RTE console command, the following status information must be provided: the time of day, the total number of active emulated remote devices, the number of suspended remote devices, and the number of remote devices suspected of having stopped.

(2) Specific device status. For a single agency-specified emulated remote device, the following information must be provided upon entry of an RTE console command: the time of day, the unique device identifier, the link to which the device is assigned, whether the device is active or suspended, the scenario that the device is using, the time of the last application data input by the device, and the time of the last application data output to the device.

PART 5. RTE MONITOR CHARACTERISTICS

16. General contents of RTE log file.

a. For each benchmark mix execution, vendors must record at least the following information in one or more RTE log files:

(1) Application data (application I/O pairs) sent and received by emulated TP devices;

(2) An indicator of the scenario that an emulated device used when application data was sent or received by that device;

(3) All device control data sent and received; e.g., cursor positioning characters, line feed and carriage return characters;

(4) All protocol data except for link control; and

(5) All messages directed to the log by scenarios.

b. At its option, a vendor may also record (in one or more RTE log files) all RTE operator console activity, including, but not limited to, all commands and responses used to suspend and restart scenarios and to obtain status information.

17. Time-stamps.

a. General. The log file must also indicate the times that certain events actually occurred. All such indicators (time-stamps) must be accurate to at least one-half second. All messages directed to the RTE log file(s) by scenarios and, when included, all RTE operator console activity must be time-stamped. Other events that must be time-stamped vary with TP device types and are described below.

b. Interactive asynchronous teleprinters and displays. The following events must be time-stamped for emulated interactive asynchronous teleprinters and interactive asynchronous displays:

(1) The start of each application I/O pair, which is either the start of the think time delay, the start of the type time delay (if no think time is defined), or the transmission to the SUT of the first character of each input line (if neither think time nor type time is defined);

(2) The transmission to the SUT of the last input character of an application I/O pair, which corresponds to the last user keystroke of each input line; e.g., carriage return;

(3) The receipt by the RTE of the first printable character of the first SUT output line resulting from the device input, or the receipt of the last non-printable character if the output line contains no printable characters; and

(4) The end of each application I/O pair, which is the receipt by the RTE of the last character of the last SUT output line resulting from the device input; e.g., line feed, prompt.

c. Interactive synchronous teleprinters and displays. For emulated interactive synchronous teleprinters and interactive synchronous displays, the following events must be time-stamped:

(1) The start of each application I/O pair, which is either the start of the think time delay, the start of the type time delay (if no think time is defined), or the point at which the emulated user would have hit the last keystroke of the input (e.g., transmit key, enter key) if neither think time nor type time is defined;

(2) The last user keystroke of the input (e.g., transmit key), which is the end of the type time delay, if defined;

(3) The receipt by the emulated device of the last character of each error-free transmission buffer of the resulting SUT output; and

(4) The end of each application I/O pair, which for these devices is either the end of any specified print time, the point at which the emulated device can accept another input keystroke (e.g., keyboard unlock command), or the receipt of the last error-free output buffer, whichever occurs last.

d. Remote batch terminals. For remote batch terminals, both the start and end of each application I/O pair must be time-stamped.

(1) Start of application I/O pair for card input. An application I/O pair begins for emulated card input when (a) the emulated device transmits the first character of a message requesting the initiation of card input, if it is the start of an input operation; (b) the previous card input delay, if defined, expires; and (c) a card input buffer becomes available in the emulated terminal; e.g., the SUT acknowledges the correct receipt of the previous card input transmission.

(2) End of application I/O pair for card input. A card input application I/O pair ends when (a) the last character of the card input block is transmitted to the SUT; and (b) the RTE receives a message from the SUT acknowledging the correct receipt of that input block, if the SUT sends an acknowledgement for every block.

(3) Start of application I/O pair for print output. An application I/O pair begins for emulated output of print files when (a) the emulated device receives the first character of a message requesting the initiation of print output, if this is the start of an output operation; (b) the last character of the previous print block (if any) is received by the RTE and the block is error-free; and (c) a print buffer becomes available in the emulated terminal; e.g., any current print delay expires, the RTE sends the SUT a positive acknowledgement for a previous block.

(4) End of application I/O pair for print output. An application I/O pair ends for print output when (a) the last character of that I/O pair's print output block is received and the block is error-free and (b) that I/O pair's print delay, if defined, expires.

e. Remote hosts. The start and end of each application I/O pair also must be time-stamped for emulated remote hosts.

(1) Start of application I/O pair for input to SUT. An application I/O pair begins for transfer of files or batch jobs from a remote host to the SUT when (a) the RTE sends the first character of a message to the SUT requesting the initiation of an input operation, if the input is just beginning; (b) the last character of the previous transmission block is sent to the SUT, if a previous block were sent; and

(c) the RTE receives a message from the SUT acknowledging the correct receipt of the previous block, if the SUT sends an acknowledgement for every block.

(2) <u>End of application I/O pair for input to SUT</u>. An application I/O pair ends for file or batch job input when (a) the last character of the I/O pair's transmission block is sent to the SUT and (b) the SUT acknowledges the correct receipt of that block, if an acknowledgement is made for every block.

(3) <u>Start of application I/O pair for output from SUT</u>. An application I/O pair begins for transfer of files or batch jobs from the SUT to the emulated remote host when (a) the RTE receives the first character of a message from the SUT requesting to initiate an output operation, if the output is just beginning; and (b) the last character of the previous transmission block is received and the block is error-free.

(4) <u>End of application I/O pair for output from SUT</u>. For file or batch job output, an I/O pair ends when the last character of that pair's output block is received and the block is error-free.

f. <u>Intermediate devices</u>. No events must be specifically time-stamped for cluster controllers, concentrators, and packet network interface devices, because these intermediate devices are configured between generic remote devices and the SUT. The time-stamps for the remote devices must be logged as defined above, when the remote devices connect either directly to the SUT or through one or more of these intermediate devices.

18. <u>Event identification</u>.

a. <u>General</u>. Vendors must be able to identify the first and last log file entries associated with certain agency-specified events. Several RTE log reports depend upon identifying these entries. (Chapter 5, part 6 describes the RTE log reports that vendors must be able to provide.)

b. <u>Scenario groups</u>. Agencies may assign each scenario to exactly one of up to 20 groups of scenarios for each single benchmark test. Agencies shall assign scenarios to groups (if any) when the scenarios are released to industry. For each occurrence of a scenario in a test, vendors must be able to identify (1) the first RTE log file entry of the first application I/O pair of that scenario, (2) the last log entry of the last application I/O pair, and (3) the group to which the scenario is assigned.

c. User functions. Agencies may select up to 20 vendor-independent user functions from all of the functions to be performed in a single benchmark test. For each occurrence of each of these functions in a test, vendors must be able to identify (1) the first RTE log entry of the first application I/O pair of that function, (2) the last entry of the last application I/O pair, and (3) the specific function that occurred. Agencies shall identify the selected functions in the scenario descriptions.

d. Application I/O pairs. Additionally, agencies may select up to 20 specific application I/O pairs from all of the pairs in a test; e.g., timesharing commands, transaction types. For each occurrence of each of these pairs, vendors must be able to identify (1) the last log entry of the application data input that began the pair, (2) the first log entry of the SUT output for that I/O pair, and (3) the specific pair that occurred. Agencies shall identify the selected I/O pairs in the scenario descriptions.

PART 6. RTE LOG ANALYSES

19. General.

a. This part describes the analyses that agencies are permitted to require vendors to perform on the RTE log file(s) created during a single benchmark test execution. Agencies may require vendors to provide one or more copies of any combination of the four reports detailed below, and/or one copy of the RTE Log Summary Tape also described. Agencies, however, shall not require any other RTE log summary reports or any modifications to the four reports or summary tape described in this part. If an agency desires additional RTE log summary data, the agency shall prepare any and all log reduction and report generation programs needed to produce the data. It is mandatory that all agency-prepared log analysis programs be in some ANSI standard language (e.g., FORTRAN, COBOL), use the RTE Log Summary Tape as input, and be fully described and distributed to vendors when the LTD documentation is released to industry. Unless explicitly stated below, agencies shall also define, when the LTD documentation is released to industry (1) the precise number, types, and parameters of all required RTE log summary reports; and (2) whether or not a copy of the RTE Log Summary Tape is required. (The RTE Log Summary Tape can be used to document the progress of the RTE portions of a benchmark mix execution, regardless of whether an agency prepares additional log analysis programs.)

b. Each vendor may choose the internal formats, media, etc. of the original log files produced by its RTE's. Multiple files may be used for a single RTE, and multiple RTE's may be used in a single benchmark test. Each vendor will ensure that all original RTE log files are processed, merged, etc. as required, to produce the agency-specified summary reports and summary tape.

c. Every page of all summary reports described below must include a procurement title, a vendor code, the date of the benchmark test execution, the page number, and the report heading. Agencies may define a procurement title of up to 40 characters and a vendor code of up to ten characters.

20. Scenario Summary Report.

a. Figure 5-20 illustrates the recommended format and required data elements of the Scenario Summary Report. Agencies will define the general time period(s) to be summarized in one or more reports; e.g., the total duration

PROCUREMENT: ________ VENDOR: ________ DATE: ________ PAGE: ________

SCENARIO SUMMARY REPORT

SUMMARY PERIOD START TIME: ________ STOP TIME: ________ DURATION: ________

SCENARIO GROUP: ____

NUMBER IN PERIOD: ____ NUMBER NOT COMPLETED: ____

NUMBER COMPLETED: ____ PERCENTAGE OF TOTAL: ____ COMPLETION RATE: ____ PER MINUTE

TURNAROUND TIME STATISTICS FOR COMPLETED SCENARIOS:

AVERAGE: ________ ______ PERCENTILE: ________

MINIMUM: ________ ______ PERCENTILE: ________

MEDIAN: ________ ______ PERCENTILE: ________

MAXIMUM: ________ ______ PERCENTILE: ________

SCENARIO GROUP: ____

°
°
°

Figure 5-20. Scenario Summary Report

of the benchmark mix execution, each third of the total mix execution duration. The precise start and stop times will be determined immediately after test completion, based upon the actual value of the RTE system clock at the start of the test. The format of the start and stop times printed in the report will be HH:MM:SS, where HH is the hour (00 through 23), MM is the minute (00-59), and SS is the second (00-59). The DURATION is the difference between the stop time and start time, and its format is also HH:MM:SS.

b. The report will provide a common set of statistics for each of up to 20 groups of scenarios and for all scenarios, regardless of group. When the scenarios are released to industry, agencies may assign each scenario to one of up to 20 groups. The scenario group name may contain up to ten characters and will be defined by the agency. "ALL" will be printed for the group name with the set of statistics that cover all scenarios, regardless of group.

c. For each scenario group, the report will provide a count of the total number of scenarios that either started or successfully ended during the period; i.e., NUMBER IN PERIOD. A scenario will have completed in a period if it both started and successfully ended in that period. A scenario will not have completed in a period if it either (1) started but did not successfully end in that period or (2) successfully ended in that period but did not start in that period. Separate counts will be provided of the number of scenarios that did complete and that did not complete. The sum of these two counts will equal the total number of scenarios in that period. The number of completed scenarios divided by the total number of scenarios in the period will be provided; i.e., PERCENTAGE OF TOTAL. The report will provide the COMPLETION RATE, defined as the number of completed scenarios divided by the total number of minutes in the summary period. The report also will include the following turnaround time statistics for completed scenarios: average, minimum, median, and maximum. In addition, agencies have the option of defining four additional percentile levels (e.g., ninety, ninety-five) that must be provided in the report. The format of these statistics will be MM:SS:H, where MM is the number of minutes, SS is the number of seconds, and H is the closest number of half-seconds (0 or 5). Scenario turnaround time is the duration between (1) the start of the first application I/O pair of a given scenario, and (2) the end of the last application I/O pair of that scenario. The specific events that comprise the start and end of an application I/O pair vary with generic device types and are described in chapter 5, part 5.

21. Function Summary Report. Figure 5-21 illustrates the recommended format and required data elements of the Function Summary Report. This report differs from the Scenerio Summary Report in only a few areas. When the scenarios are released to industry, agencies may select up to 20 vendor-independent user functions from all the functions described. The Function Summary Report will provide a common set of statistics for each of these functions, and for all the selected functions in the summary period. The function name will be defined by the agency and may contain up to ten characters. The function summary statistics are defined as the scenario statistics were defined above, except that (a) the function statistics are based on the number of functions in the summary period and (b) the COMPLETION RATE is in functions per second.

22. Interactive Response Time Summary Report.

a. Figure 5-22 illustrates the recommended format and required data elements of the Interactive Response Time Summary Report. This report is defined for, and thus can be used with, only interactive scenarios; i.e., scenarios assigned to interactive devices. This report also differs from the Scenario Summary Report in a few areas. When the scenarios are released to industry, agencies may select up to 20 specific application I/O pairs from all of the pairs in the interactive scenarios; e.g., timesharing commands, single-input transactions. The Interactive Response Time Summary Report will provide a set of statistics for each of these I/O pairs, and for all application I/O pairs in all the interactive scenarios in the test. Agencies may define a name of up to ten characters for each selected I/O pair. The response time summary statistics are defined as the scenario statistics were defined, except that (1) the statistics are based on the occurrences of application I/O pairs during the period, (2) the COMPLETION RATE is in I/O pairs per second, and (3) response time (not turnaround time) is the primary measure. The response time definitions given below (and repeated elsewhere in this handbook) differ from the definition in FIPS PUB 57, "Guidelines for the Measurement of Interactive Computer Service Turnaround Time and Response Time." Agencies shall use the definitions in this handbook during acquisitions using remote terminal emulation.

b. For interactive asynchronous teleprinters and interactive asynchronous displays, response time is the elasped time between (1) the transmission to the SUT of the last input character of a given I/O pair which corresponds

PROCUREMENT: ________ VENDOR: ________ DATE: ________ PAGE: ________

FUNCTION SUMMARY REPORT

SUMMARY PERIOD START TIME: ________ STOP TIME: ________ DURATION: ________

FUNCTION: _____

NUMBER IN PERIOD: ____ NUMBER NOT COMPLETED: ____

NUMBER COMPLETED: ____ PERCENTAGE OF TOTAL: ____ COMPLETION RATE: ____ PER SECOND

TURNAROUND TIME STATISTICS FOR COMPLETED FUNCTIONS:

AVERAGE: ________ _____ PERCENTILE: ________

MINIMUM: ________ _____ PERCENTILE ________

MEDIAN: ________ _____ PERCENTILE: ________

MAXIMUM: ________ _____ PERCENTILE: ________

FUNCTION: _____

∘
∘
∘

Figure 5-21. Function Summary Report

PROCUREMENT: ________ VENDOR: ________ DATE: ________ PAGE: ________

INTERACTIVE RESPONSE TIME SUMMARY REPORT

SUMMARY PERIOD START TIME: ________ STOP TIME: ________ DURATION: ________

APPLICATION INPUT-OUTPUT PAIR: ____

NUMBER IN PERIOD: ____ NUMBER NOT COMPLETED: ____

NUMBER COMPLETED: ____ PERCENTAGE OF TOTAL: ____ COMPLETION RATE: ____ PER SECOND

RESPONSE TIME STATISTICS FOR COMPLETED PAIRS:

AVERAGE: ________ ____ PERCENTILE: ________

MINIMUM: ________ ____ PERCENTILE: ________

MEDIAN: ________ ____ PERCENTILE: ________

MAXIMUM: ________ ____ PERCENTILE: ________

APPLICATION INPUT-OUTPUT PAIR: ____

o
o
o

Figure 5-22. Interactive Response Time Summary Report to the last keystroke of an emulated user, and (2) either the receipt by the RTE of the first printable character of the first SUT output line of the resulting output, or the receipt of the last non-printable character if the resulting output contains no printable characters.

c. For interactive synchronous teleprinter and interactive synchronous displays, response time is the elasped time between (1) the last user keystroke of the input and (2) the receipt by the RTE of the last character of the first error-free transmission buffer of the resulting output.

23. RTE Log Summary Report.

a. General. The required data elements of the RTE Log Summary Report are described below. A specific report format is not recommended because of the high variability of possible data values and field lengths between vendors. As described above for the other log reports, however, each page of this report also should contain a procurement title, a vendor code, the date of the benchmark test execution, the page number, and the report heading. The report entries should be printed in order of occurrence; i.e., increasing time-stamps.

b. RTE operator console activity. If a vendor chooses to record RTE operator console activity in the RTE log file, the following data elements must be included in the RTE Log Summay Report for RTE-related data entered, printed, or displayed on the RTE operator console:

(1) An indicator of whether the data was entered or output on the console;

(2) The time that the data was entered or output; and

(3) The RTE-related data, including but not limited to all operator inputs and all RTE system responses.

c. RTE log messages. For each message directed to the RTE log file by a scenario, the following data elements must be included in the report:

(1) The agency-specified name of the scenario that initiated the message;

(2) The unique identifier of the emulated remote device using that scenario;

(3) The unique identifier of the data communication link to which that remote device was assigned;

(4) An indicator of the generic type and/or make and model of that remote device;

(5) An indicator that the message was directed to the log file by a scenario and was not a transmission to or from the SUT;

(6) The time that the scenario directed the message to the log file; and

(7) The alphanumeric text of the message.

d. Input and output transmissions. The following data elements must be included in the RTE Log Summary Report for each transmission (asynchronous line or synchronous block) sent or received by an emulated remote device:

(1) The unique identifier of the emulated remote device that sent or received the transmission;

(2) The unique identifier of the data communication link to which that device was assigned;

(3) An indicator of the generic type and/or make and model of that remote device;

(4) An indicator of whether the transmission was sent or received by the device;

(5) The agency-specified name of the scenario used by that device when the transmission occurred;

(6) An indicator of whether this transmission was either the first or last log entry of an agency-specified event; e.g., start of a scenario, end of a user function (The possible events are detailed in chapter 5, part 5.);

(7) The total size of the transmission in alphanumeric characters;

(8) All associated time-stamps (See chapter 5, part 5.);

(9) The transmission text, printed in alphanumeric characters; non-printable data (e.g., line feed, cursor

control) will be shown as a vendor-chosen, standard default character; e.g., period, space; and

(10) The transmission text printed in hexadecimal.

e. Local printing of display contents. Each time the display contents of a synchronous interactive display are printed without any data transmissions to and from the SUT, the report must include the following data elements:

(1) The unique identifier of the emulated remote device that initiated the printing;

(2) The unique identifier of the data communication link to which that device was assigned;

(3) The agency-specified name of the scenario used by that device when the printing occurred;

(4) An indicator of whether this printing resulted in either the first or last log entry of an agency-specified event; e.g., end of a user function (The possible events are detailed in chapter 5, part 5.);

(5) The total amount of data printed in alphanumeric characters;

(6) All associated time-stamps (See chapter 5, part 5.); and

(7) The printed data.

f. Report options. Agencies may specify the required number of RTE Log Summary Reports and the summary options for each, any time before proposal due date and/or after the completion of a benchmark mix execution. For each report, agencies shall specify the time period to be summarized and one of the following summary options: (1) RTE Console Activity, (2) Scenario Log Messages, (3) Link xxx and Device yyy, (4) Link xxx, and (5) All. When the RTE Console Activity option is selected, and if a vendor chooses to record this information, the resulting report will include only those data elements associated with RTE operator console activity. For the Scenario Log Messages option, the resulting report will include only the data elements required for messages directed to the log file by scenarios. For the third option, the resulting report will include all data elements associated with (1) all transmissions sent or received by device yyy on line xxx, (2) all display print operations performed by device yyy on line xxx, (3) all log file messages initiated by scenarios on device yyy on line xxx, and (4) all RTE console activity (if recorded). When an agency specifies the Link xxx summary option, the resulting report will include the same data elements as with the previous option, except that all activity by the devices on the specified link will be included. For the last summary option, all activity on all links and devices and on the RTE console (if recorded) will be included.

24. RTE Log Summary Tape.

a. General.

(1) Agencies have the option of requiring vendors to provide, for each benchmark test execution, a summary of the RTE log file(s) on tape in the standard format specified below. The physical tape characteristics, data elements and logical record formats for the summary tape are detailed below. All of the figures referenced in paragraph 24 are located at the end of this paragraph.

(2) Except for the Log-Header and End-Log-Data records described below, log summary records on the tape will be in the order of (a) increasing link identifier, (b) increasing device identifiers, (c) increasing TIME1 values, and (d) increasing TIME2 values. The result of this ordering is that all log data for a single device will be stored together on the tape in time sequence, and that the log data for all devices on a single link will be grouped together.

b. Physical tape characteristics. The physical characteristics of the RTE Log Summary Tape are as follows:

(1) 8-bit ASCII character set (See ANSI X3.4-1977.);

(2) ANSI standard tape label, multi-volume (if needed), single-file format, complying with either level 2 or level 4 of ANSI X3.27-1975;

(3) ANSI standard 9-track, 1600 characters per inch (cpi) or 6250 cpi format (agency option) (See ANSI X3.39-1973 or X3.54-1975.);

(4) Variable length logical records spanning fixed, 2048-character length physical blocks (See ANSI X3.27-1975.);

(5) Minimum physical block length of 18 characters; filled (if necessary) with any legal data characters.

c. Logical record types. The RTE Log Summary Tape will contain up to eight logical record types. Some data element definitions vary with logical record types. The logical record types and associated data element definitions are presented below.

(1) Log-Header record. Figure 5-24.1 defines the format of the Log-Header logical record type. A single Log-Header record will be included in each tape and will be the first data record on the tape. The LOG-REC-TYPE value for the record is "LHDR." The PROCUREMENT-TITLE and VENDOR-CODE values may be supplied by the procuring agencies. The DATE value is the actual YEAR, MONTH, and DAY that the specific benchmark test execution occurred. The REMOTE-DEVICE-COUNT value is the total number of remote devices emulated for the benchmark test. The value of LINK-DEVICE-CONFIGURATION is a table detailing the assignment of devices to links, link speeds, etc. for the test. There is one row in the table for every emulated remote device in the test. The value of LINK-NUMBER is the unique identifier (on the RTE system) of the data communication link to which the associated emulated remote device is connected. The LINK-SPEED value is the speed of the link, in bits per second. The value of LINK-TYPE is "FDX" if the link is full-duplex, or "HDX" if the link is half-duplex. The value of REMOTE-DEVICE-NUMBER is the unique identifier (on the RTE system) of the emulated remote device. The REMOTE-DEVICE-TYPE value is the abbreviation, from figure 5-24.2, for the type of the associated remote device. The value of INTERMEDIATE-DEVICE-NUMBER is the unique identifier (on the RTE system) of any intermediate device configured between the associated remote device and the SUT; if no intermediate device were configured, the value is four blank characters. The INTERMEDIATE-DEVICE-TYPE value is the abbreviation, from figure 5-24.2, of the type of any intermediate device configured between the associated remote device and the SUT; the value is two blank characters if no intermediate device were configured. The value of RTE-NUMBER is a unique identifier of the RTE used to emulate the remote device. The rows of the LINK-DEVICE-CONFIGURATION table will be ordered by increasing values of (a) RTE-NUMBER, (b) LINK-NUMBER, and (c) REMOTE-DEVICE-NUMBER.

(2) Device-Transmit record. The format of the Device-Transmit logical record type is detailed in figure 5-24.3. A single record of this type will be stored on the

summary tape for each input transmission of application data from an emulated remote device to the SUT. For devices using asynchronous protocols, an input transmission is a single input line of data. For devices using synchronous protocols, an input transmission is a single input block of data. The LOG-REC-TYPE value for this type of record is "DXMT." The DEVICE-TYPE value depends upon the generic type of the emulated remote device, and the required abbreviations are given in figure 5-24.2. The value of LINK-NUMBER is the unique identifier (on the RTE system) of the data communication link to which the emulated remote device is connected. The DEVICE-NUMBER value is the unique identifier (on the RTE system) of the emulated remote device. The definitions of the TIME1 and TIME2 values vary with the remote device type, and are presented in figure 5-24.4. The value of the SCENARIO-NAME is the agency-specified name of the scenario used by the emulated device when the transmission occurred. The EVENT-FLAG is used to indicate the first and last log records associated with certain agency-specified events (see chapter 5, part 5). The value of EVENT-FLAG in a given record is four blank characters unless that record corresponds to an agency-specified event. The required EVENT-FLAG values for possible events are defined in figure 5-24.5. The MESSAGE-SIZE value is the size in characters of the alphanumeric equivalent of the input transmission. The value of ALPHA-MESSAGE-TEXT is the alphanumeric equivalent of the input transmission, with all non-printable characters represented by a blank character. The value of HEX-MESSAGE-TEXT is the alphanumeric representation of the hexidecimal equivalent of the input transmission.

(3) Device-Receive record. The format of the Device-Receive logical record type is shown in figure 5-24.6. A single record of this type will be stored on the summary tape for each output transmission of application data from the SUT to an emulated remote device. For devices using asynchronous protocols, an output transmission is a single line of output data. For devices using synchronous protocols, an output transmission is a single output block of data. The LOG-REC-TYPE value for this record type is "DRCV." The definitions of the TIME1 and TIME2 values vary with the remote device type and are detailed in figure 5-24.7. All other data element definitions for this record type are the same as for the Device-Transmit record.

(4) Device-Print record. Figure 5-24.8 presents the format of the logical record type Device-Print. A single record of this type will be stored on the summary

tape each time the display contents of an interactive synchronous display are printed on an attached character printer, and the print operation does not result in data transmissions to or from the SUT. The LOG-REC-TYPE value is "DPRT," and the DEVICE-TYPE value is "SD." TIME1 is defined as the start time of the application I/O pair, which is either the start of the think time delay, the start of the type time delay (if no think time is defined), or the point at which the emulated user would have hit the key that initiated the print operation, if neither think time nor type time is defined. TIME2 is defined as the end time of the I/O pair, which is either the expiration of the print time or the point at which the emulated device can accept another input keystroke, whichever occurs last. The value of ALPHA-MESSAGE-TEXT is the alphanumeric equivalent of the display contents that was to be printed.

(5) RTE-Console-Input and RTE-Console-Output records. The formats of the RTE-Console-Input and the RTE-Console-Output logical record types are shown in figures 5-24.9 and 5-24.10, respectively. (Each vendor may choose to either document all RTE operator console activity in the RTE Log Summary Tape, or provide agencies printed copies of all RTE console activity after the completion of each benchmark test execution.) The LOG-REC-TYPE value for the first record is "CNLI," and the value for the second record is "CNLO." For both these record types, the value of LINK-NUMBER is "999," and the value of DEVICE-NUMBER is "9999." For the RTE-Console-Input record type, TIME1 is the time that the RTE operator entered data at the RTE operator console. For the RTE-Console-Output record type, TIME1 is the time that data were output on the RTE operator console. For both record types, the value of TIME2 is zero and the value of MESSAGE-SIZE is the number of characters in ALPHA-MESSAGE-TEXT. ALPHA-MESSAGE-TEXT is the alphanumeric text of either the console input or output, as appropriate.

(6) Scenario-Message record. Figure 5-24.11 illustrates the format of the Scenario-Message logical record type. A single record of this type will be stored on the summary tape each time a scenario directs a message to the RTE log file. The value of LOG-REC-TYPE is "INFO." The value of TIME1 is the time that the scenario initiated the message, and the value of TIME2 is zero. All other data fields are defined as they are for the Device-Transmit record type.

(7) End-Log-Data record. Figure 5-24.12 details the format of the End-Log-Data logical record type. One record of this type will be included in each tape and will be the last data record on the tape. The value of LOG-REC-TYPE is "ENDD."

```
01  LOG-HEADER.

    02  LOG-REC-TYPE                         PIC X(4).
    02  PROCUREMENT-TITLE                    PIC X(40).
    02  VENDOR-CODE                          PIC X(10).
    02  DATE.

        03  YEAR                             PIC X(2).
        03  MONTH                            PIC X(2).
        03  DAY                              PIC X(2).

    02  REMOTE-DEVICE-COUNT                  PIC X(4).
    02  LINK-DEVICE-CONFIGURATION OCCURS
        1 TO 4000 TIMES DEPENDING ON
        REMOTE-DEVICE-COUNT.

        03  LINK-NUMBER                      PIC X(3).
        03  LINK-SPEED                       PIC X(5).
        03  LINK-TYPE                        PIC X(3).
        03  REMOTE-DEVICE-NUMBER             PIC X(4).
        03  REMOTE-DEVICE-TYPE               PIC X(2).
        03  INTERMEDIATE-DEVICE-NUMBER       PIC X(4).
        03  INTERMEDIATE-DEVICE-TYPE         PIC X(2).
        03  RTE-NUMBER                       PIC X.
```

Figure 5-24.1. COBOL data definition for logical record type Log-Header

TP DEVICE TYPE	ABBREVIATION
Interactive Asynchronous Teleprinter	AT
Interactive Asynchronous Display	AD
Interactive Synchronous Teleprinter	ST
Interactive Synchronous Display	SD
Remote Batch Terminal	RB
Remote Host System	RH
Cluster Controller	CC
Concentrator	CN
Packet Network Interface Device	PK
Other	OH

Figure 5-24.2. TP device type abbreviations

```
01  DEVICE-TRANSMIT.

    02  LOG-REC-TYPE                      PIC X(4).
    02  DEVICE-TYPE                       PIC X(2).
    02  LINK-NUMBER                       PIC X(3).
    02  DEVICE-NUMBER                     PIC X(4).
    02  TIME1.

        03  HOUR                          PIC 9(2).
        03  MINUTE                        PIC 9(2).
        03  SECOND                        PIC 99.9.

    02  TIME2.

        03  HOUR                          PIC 9(2).
        03  MINUTE                        PIC 9(2).
        03  SECOND                        PIC 99.9.

    02  SCENARIO-NAME                     PIC X(10).
    02  EVENT-FLAG.

        03  EVENT-TYPE                    PIC X(2).
        03  EVENT-NUMBER                  PIC X(2).

    02  MESSAGE-SIZE                      PIC 9(4).
    02  MESSAGE-TEXT OCCURS
        1 TO 4000 TIMES DEPENDING
        ON MESSAGE-SIZE.

        03 ALPHA-MESSAGE-TEXT             PIC X.
        03 HEX-MESSAGE-TEXT               PIC X(2).
```

Figure 5-24.3. COBOL data definition for logical record type Device-Transmit

DEVICE-TYPE VALUES	DATA FIELD NAME	DEFINITION
AT and AD	TIME1	The time of the start of an application I/O pair, which is either the start of the think time delay, the start of the type time delay (if no think time is defined), or the transmission to the SUT of the first character of the input line (if neither think time nor type time is defined)
	TIME2	The time of the transmission to the SUT of the last input character of an application I/O pair (e.g., carriage return), which corresponds to the last user keystroke of the input line
ST and SD	TIME1	The time of the start of an application I/O pair, which is either the start of the think time delay, the start of the type time delay (if no think time is defined), or the point at which the emulated user would have hit the last keystroke of the input (e.g., transmit key, enter key) if neither think time nor type time are defined
	TIME2	The time at which the emulated user would have hit the last keystroke of the input; e.g., transmit key

Figure 5-24.4. Time-stamp definitions for logical record type
(Part 1 of 3) Device-Transmit

DEVICE-TYPE VALUES	DATA FIELD NAME	DEFINITION
RB	TIME1	The time of the start of an application I/O pair, which occurs when (1) the emulated device transmits the first character of a message requesting the initiation of card input, if it is the start of an input operation; (2) the previous card input delay, if defined, expires; and (3) a card input buffer becomes available in the emulated terminal; e.g., the SUT acknowledges the correct receipt of the previous card input transmission
	TIME2	The time of the end of an application I/O pair, which is when (1) the last character of the input block is transmitted to the SUT, and (2) the RTE receives a message from the SUT acknowledging the correct receipt of that input block, if the SUT sends an acknowledgment for every block
RH	TIME1	The time of the start of an application I/O pair, which occurs when (1) the RTE sends the first character of a message to

Figure 5-24.4. Time-stamp definitions for logical record (Part 2 of 3) type Device-Transmit

DEVICE-TYPE VALUES	DATA FIELD NAME	DEFINITION
RH (Cont'd)		the SUT requesting the initiation of an input operation, if the input is just beginning; (2) the last character of the previous transmission block is sent to the SUT, if a previous block was sent; and (3) the RTE receives a message from the SUT acknowledging the correct receipt of the previous block, if the SUT sends an acknowledgment for every block
	TIME2	The time of the end of an application I/O pair, which is when (1) the last character of the input transmission block is sent to the SUT, and (2) the SUT acknowledges the correct receipt of that block, if an acknowledgment is made for every block

Figure 5-24.4. Time-Stamp definitions for logical record
(Part 3 of 3) type Device-Transmit

EVENT-FLAG SUBFIELD: EVENT-TYPE VALUE	EVENT-FLAG SUBFIELD: EVENT-NUMBER VALUE	EVENT DEFINITION
SF	x	The first log record of the first application I/O pair of a scenario assigned to group x, where $0 \leq x \leq 19$
[illegible]	x	The last log record of the last application I/O pair of a scenario assigned to scenario group x, where $0 \leq x \leq 19$
FF	x	The first log record of the first application I/O pair of user function x, where $0 \leq x \leq 19$
FL	x	The last log record of the last application I/O pair of user function x, where $0 \leq x \leq 19$
FO	x	The only log record of user function x, where $0 \leq x \leq 19$
PF	x	The last log record of the application data input that began application I/O pair x, where $0 \leq x \leq 19$
PL	x	The first log record of the SUT output for application I/O pair x, where $0 \leq x \leq 19$
PO	x	The only log record of the application I/O pair x, where $0 \leq x \leq 19$

Figure 5-24.5. EVENT-FLAG definitions

```
01  DEVICE-RECEIVE.

    02  LOG-REC-TYPE                  PIC X(4).
    02  DEVICE-TYPE                   PIC X(2).
    02  LINK-NUMBER                   PIC X(3).
    02  DEVICE-NUMBER                 PIC X(4).
    02  TIME1.

        03  HOUR                      PIC 9(2).
        03  MINUTE                    PIC 9(2).
        03  SECOND                    PIC 99.9.

    02  TIME2.

        03  HOUR                      PIC 9(2).
        03  MINUTE                    PIC 9(2).
        03  SECOND                    PIC 99.9.

    02  SCENARIO-NAME                 PIC X(10).
    02  EVENT-FLAG.

        03  EVENT-TYPE                PIC X(2).
        03  EVENT-NUMBER              PIC X(2).

    02  MESSAGE-SIZE                  PIC 9(4).
    02  MESSAGE-TEXT OCCURS
        1 to 4000 TIMES DEPENDING
        ON MESSAGE-SIZE.

        03  ALPHA-MESSAGE-TEXT        PIC X.
        03  HEX-MESSAGE-TEXT          PIC X(2).
```

Figure 5-24.6. COBOL data definition for logical record type Device-Receive

DEVICE-TYPE VALUES	DATA FIELD NAME	DEFINITION
AT and AD	TIME1	For the <u>first</u> DEVICE-RECEIVE log record of an application I/O pair, the time of either (1) the receipt by the RTE of the first printable character of the first SUT output line; or (2) the receipt of the last non-printable character of the output, if the output contains no printable characters
		For all but the first DEVICE-RECEIVE log record of an application I/O pair, the value is zero
	TIME2	For the <u>last</u> DEVICE-RECEIVE log record of an application I/O pair, the time of the receipt by the RTE of the last character of the SUT output line; e.g., line feed, prompt
		For all but the last DEVICE-RECEIVE log record of an application I/O pair, the value is zero
ST and SD	TIME1	The time of the receipt by the emulated device of the last character of an error-free transmission block of the resulting SUT output
	TIME2	The time of either the expiration of any specified print time, the receipt of the last character of an error-free output block, or the point at which the emulated device can accept another input keystroke (e.g., keyboard unlock), whichever occurs last

Figure 5-24.7. (Part 1 of 3) Time-stamp definitions for logical record type Device-Receive

DEVICE-TYPE VALUES	DATA FIELD NAME	DEFINITION
RB	TIME1	The time of the start of an application I/O pair, which is when (1) the emulated device receives the first character of a message requesting the initiation of print output, if this is the start of an output operation; (2) the last character of the previous output block (if any) is received by the RTE and the block is error-free; and (3) a print buffer becomes available in the emulated terminal; e.g., any current print delay expires, the RTE sends the SUT a positive acknowledgement for a previous block.
	TIME2	The time of the end of an application I/O pair, which is when (1) the last character of the output block is received and the block is error-free, and (2) any defined print delay expires
RH	TIME1	The time of the start of an application I/O pair, which is when (1) the RTE receives the first character of a message from the SUT requesting the initiation of an output operation, if the output is just beginning; and (2) the last character of the previous transmission block is received and the block is error-free

Figure 5-24.7. Time-stamp definitions for logical record type Device-Receive
(Part 2 of 3)

DEVICE-TYPE VALUES	DATA FIELD NAME	DEFINITION
RH (Cont'd)	TIME2	The time of the end of an application I/O pair, which is when the last character of the output block is received and the block is error-free

Figure 5-24.7. (Part 3 of 3) Time-stamp definitions for logical record type Device-Receive

```
01  DEVICE-PRINT.

    02  LOG-REC-TYPE                        PIC X(4).
    02  DEVICE-TYPE                         PIC X(2).
    02  LINK-NUMBER                         PIC X(3).
    02  DEVICE-NUMBER                       PIC X(4).
    02  TIME1.

        03  HOUR                            PIC 9(2).
        03  MINUTE                          PIC 9(2).
        03  SECOND                          PIC 99.9.

    02  TIME2.

        03  HOUR                            PIC 9(2).
        03  MINUTE                          PIC 9(2).
        03  SECOND                          PIC 99.9.

    02  SCENARIO-NAME                       PIC X(10).
    02  EVENT-FLAG.

        03  EVENT-TYPE                      PIC X(2).
        03  EVENT-NUMBER                    PIC X(2).

    02  MESSAGE-SIZE                        PIC 9(4).
    02  ALPHA-MESSAGE-TEXT OCCURS
        1 to 4000 TIMES DEPENDING
        ON MESSAGE-SIZE                     PIC X.
```

Figure 5-24.8. COBOL data definition for logical record type Device-Print

```
01  RTE-CONSOLE-INPUT.

    02  LOG-REC-TYPE                      PIC X(4).
    02  LINK-NUMBER                       PIC X(3).
    02  DEVICE-NUMBER                     PIC X(4).
    02  TIME1.

        03  HOUR                          PIC 9(2).
        03  MINUTE                        PIC 9(2).
        03  SECOND                        PIC 99.9.

    02  TIME2.

        03  HOUR                          PIC 9(2).
        03  MINUTE                        PIC 9(2).
        03  SECOND                        PIC 99.9.

    02  ALPHA-MESSAGE-TEXT                PIC X(140).
```

Figure 5-24.9. COBOL data definition for logical record type RTE-Console-Input

```
01  RTE-CONSOLE-OUTPUT.

    02  LOG-REC-TYPE                        PIC X(4).
    02  LINK-NUMBER                         PIC X(3).
    02  DEVICE-NUMBER                       PIC X(4).
    02  TIME1.

        03  HOUR                            PIC 9(2).
        03  MINUTE                          PIC 9(2).
        03  SECOND                          PIC 99.9.

    02  TIME2.

        03  HOUR                            PIC 9(2).
        03  MINUTE                          PIC 9(2).
        03  SECOND                          PIC 99.9.

    02  MESSAGE-SIZE                        PIC 9(4)
    02  ALPHA-MESSAGE-TEXT OCCURS 1
        TO 4000 TIMES DEPENDING ON
        MESSAGE-SIZE                        PIC X.
```

Figure 5-24.10. COBOL data definition for logical record type RTE-Console-Output

```
01  SCENARIO-MESSAGE.

    02  LOG-REC-TYPE                      PIC X(4).
    02  DEVICE-TYPE                       PIC X(2).
    02  LINK-NUMBER                       PIC X(3).
    02  DEVICE-NUMBER                     PIC X(4).
    02  TIME1.

        03  HOUR                          PIC 9(2).
        03  MINUTE                        PIC 9(2).
        03  SECOND                        PIC 99.9.

    02  TIME2.

        03  HOUR                          PIC 9(2).
        03  MINUTE                        PIC 9(2).
        03  SECOND                        PIC 99.9.

    02  SCENARIO-NAME                     PIC X(10).
    02  ALPHA-MESSSAGE-TEXT               PIC X(40).
```

Figure 5-24.11. COBOL data definition for logical record type Scenario-Message

```
01  END-LOG-DATA.

    02  LOG-REC-TYPE                      PIC X(4).
```

Figure 5-24.12. COBOL data definition for logical record type End-Log-Data

MANDATORY PROVISIONS

Chapter	Page	Paragraph
4	11	6.b.(4)
4	13	6.c.(5)
4	14	6.c.(7)
4	15	6.c.(7)
4	17	8.b.
4	18	8.c.
4	19	8.d.
4	19	8.d.(4)
4	22	8.h.
4	27	11.a.(1)
4	28	11.a.(2)
4	31	11.c.(4)
4	31	11.c.(5)
4	34	12.a.
4	34	12.c.
4	43	14.d.(4)
4	48	18.
4	50	20.a.
4	51	20.d.
4	52	21.
5	1	1.b.
5	2	2.
5	3 & 4	3.b.
5	6	6.a.
5	6	6.b.
5	7	7.a.
5	7	7.b.
5	7	7.c.
5	7	7.d.
5	7	7.e.(3)
5	22	10.a.
5	23 & 24	10.c.
5	26	12.a.
5	26	12.b.
5	27	13.b.
5	28	13.d.
5	29	14.c.(1)
5	29	14.c.(2)
5	30	14.c.(3)
	36	18.b.
5	37 & 38	18.c.
5	37 & 38	18.d.
5	39	19.a.
5	42	22.a.
5	47	23.f.

Appendix A. Reference List of Mandatory Provisions of the Handbook

SCENARIO IMPLEMENTATION INSTRUCTIONS

In the implementation of the Text Edit Scenario the following instructions must be followed:

1. The scenario consists of sequential steps, which define in English the workload demands to be performed. The steps must be performed in sequence.

2. Each numbered English language step must be represented by at least one vendor command language entry, unless otherwise specified.

3. The word "List" is used to indicate that a specific set of lines should be printed at the terminal.

4. The words "Find and print..." and "...and print the changed lines" do not imply the necessity of a separately issued print command. The use of a verify feature is acceptable.

5. The remaining portion of those lines which are larger than the carriage width of the terminal being emulated must be printed on the line immediately following.

6. The delay time, in seconds, stated for each step is the sum of the think time and type time for that step. Alternately, this is the time between the appearance of the system prompt character at the emulated terminal and the sending of the last character from the emulated terminal to the proposed configuration. (This definition of delay time is only accurate for interactive asynchronous remote devices.) The vendor RTE must implement these minimum delays in the script. For steps which require multiple lines of input, the delay times include think time and type time for all lines. The vendor may divide the delay between the multiple lines.

7. Line numbers may either be entered by the emulated terminal as part of the text to be inserted or supplied by the text editor in an auto-prompt mode.

8. All phrases in the English language steps, which are contained in quotes, are references to the contents of the edited file. Also, all references to the contents of the edited file are contained in quotes.

Appendix B. Example Scenario with Dialogue and Implementation Instructions

9. If the proposed editor does not have the facility to perform the functions indicated in step 44, it is acceptable to make a copy of the source file to be edited prior to step 2 and then perform all indicated operations on the copied file. The sample dialogue uses this second approach. In either case, the copy of the file must be made during, and by, the execution of the scenario.

10. Both upper and lower case characters must be used.

11. If line numbers are used, the capability to print the text both with and without line numbers is required; lines must be printed with line numbers unless otherwise indicated.

12. In developing this scenario, extra quotes were required which sometimes caused misalignment of columns. The vendor must make sure the file prints as in the attached sample dialogue. Extra characters must be inserted or deleted as needed.

13. Whenever "rearrange" is used, the length of the lines must be adjusted so they are as long as possible without exceeding the specified maximum line length. The specified length applies to the text only, and not to the line numbers. All lines must begin in column 1. Lines should be broken only on a blank. Blank lines should be used as delimiters between lines to be realigned.

14. The execution of the scenario requires multiple copies of TE, the text edit file, to be stored on-line.

15. The vendor should use the attached sample dialogue, implemented on an IBM System 360 (Model 65) using WYLBUR, for guidance. In all cases, however, the English language steps take precedence over the sample dialogue.

TEXT EDIT (E)

The following steps represent the Text Edit Scenario:

1. LOGON. Delay = 20.

2. Access file ______ for editing. The name of this file is TE, e.g., TE001. Delay = 12.

3. Print the entire file. Delay = 3.

4. (a) Rearrange the 54th through 64th lines such that no line is more than 37 characters long.
(b) List the new lines. Delay = 22.

5. In the line containing "least-errors", change "-" to "-b̸" and print the changed line. Delay = 23.

6. (a) Rearrange five lines beginning with the line containing "Syntax-" such that no line is more than 35 characters long.
(b) List these new lines. Delay = 22.

7. Copy the 6th through 14th lines to the end of the file. Delay = 17.

8. Delete columns 42 through 80 of the 6th through 23rd lines and print the changed lines. Delay = 23.

9. Delete columns 1 through 41 of the last nine lines and print the changed lines. The old contents of columns 42 through 80 should now be in columns 1 through 39. Delay = 23.

10. In the 12th line, change "-" to "-b̸" and print the changed line. Delay = 21.

11. Rearrange the 11th through 13th lines so no line is more than 37 characters long and print the changed lines. Delay = 22.

12. (a) Print the 6th through 23rd and last nine lines.
(b) Make all changes needed to make sure the lines print as in the last print of step 12 in the attached sample dialogue. Delay = 13.

13. Move the last nine lines from the bottom of file to the space between the 36th and 37th lines of the file. Delay = 18.

14. Indent the 6th through 45th lines so all lines start in column 2. Delay = 37.

15. Move the 46th through 59th lines to the end of the file. Delay = 19.

16. Indent the 49th through 65th lines so all lines start in column 2. Delay = 37.

17. List the entire file. Delay = 3.

18. In any case, where "pp." is on a separate line from the actual page numbers, #, the two lines should be changed so the entire phrase "pp.bb#-#" is on the second line. Maximum line length of 38 and any indentations must be maintained. Print the changed lines. Delay = 33.

19. Find and print the line containing "Sadowski". Delay = 10.

20. List all lines from that line to the end of the file. There should be 14 lines. Delay = 7.

21. (a) Copy, line by line, the last 14 lines into columns 42 through 80 of lines 2 through 15. This should result in a double column page.
(b) Delete the last 14 lines of the file. Delay = 39.

22. (a) List the first 16 lines.
(b) Make all changes needed to make sure the lines print as in the last print of step 22 in the attached sample dialogue. Delay = 8.

23. List the entire file unnumbered with the following conditions:
(a) Manual intervention (e.g., insertion of fresh paper, typing of text in local mode) should be possible at both the top and bottom of the listed text.
(b) No other text, e.g., command prompts, may be printed within two inches from the top or bottom of the text. Delay = 18.

24. Renumber the file. Delay = 5.

25. Save the edited file under the particular execution of file name E1, e.g., F1001. Delay = 12.

26. LOGOFF. Delay = 7.

```
STEP 1      LOGON FE93
            FE93 LOGON IN PROGRESS

            ENTER OPTIONAL ACCOUNT CODE

            ENTER PROGRAMMER NAME

            DATA SET SETUP.CLIST NOT IN CATALOG
            READY
SEE NOTE 2  TERMINAL LINESIZE(80)
            READY
STEP 2      WYLALLOC
            READY
            WYLBUR
            *** TSOFMT *** FORMATTING   399 PAGES ON FILE 0
            *** FORMATTING SUCCESSFULLY COMPLETED ***

            COMMAND? SET UPLOW
            COMMAND? SET LENGTH 80
            COMMAND? USE TE001
            VOLUME IS USER02
STEP 3      COMMAND? L
                1.     $
                2.     SOFTWARE ENGINEERING (SEE ALSO
                3.     PROGRAMMING LANGUAGES AND TESTING,
                4.     VALIDATING AND AUDITING)
                5.
                6.     DAVIS, R. M., "QUALITY SOFTWARE CAN      MILLER, E. F., JR. AND LIND
            AMOOD,
                7.     CHANGE THE COMPUTER INDUSTRY, "PROC.     G. E., "STRUCTURED PROGRAMM
            ING:
                8.     OF THE SYMPOSIUM ON COMPUTER PROGRAM     TOP-DOWN APPROACH," DATAMAT
            ION 19,
                9.     TEST METHODS, CHAPEL HILL, N.C., JUNE    NO. 12, DEC.  1973 PP. 12-23
```

Appendix B

STEP 3
(Continued)

```
   10.     21-23, 1972; ALSO IN PROGRAM TEST
   11.     METHODS, W. C. HETZEL, ED.                STEWART, S. L., "STAPLE, AN

   12.     (PRENTICE-HALL, INC., ENGLEWOOD           EXPERIMENTAL STRUCTURED PROG
RAMMING
   13.     CLIFFS, N.J., 1973), PP.  303-311.        LANGUAGE," COMPUT.  LANGUAG
ES 1, NO.
   14.                                               1, JAN. 1975, PP. 61-71.
   15.     HUGHES, C. E.  AND WALKER, J. C.,
   16.     "POPSS, A SYSTEM FOR MODELLING AND        SADOWSKI, W. L. AND LOZIER,
 D. W.,
   17.     ANALYZING OPERATING SYSTEM RESOURCE       "A UNIFIED STANDARDS APPROC
H TO
   18.     ALLOCATION STRATEGIES," PROC. OF THE      ALGORITHM TESTING," PROC. A
CM
   19.     3RD TEXAS CONFERENCE ON COMPUTING         SIGPLAN SYMP. ON COMPUTER PR
OGRAM
   20.     SYSTEMS, AUSTIN, TEX., NOV. 7-8,          TEST METHODS, CHAPEL HILL, N
.C., JUNE
   21.     1974, PAPER 74-CH0995-3C (IEEE            21-23, 1972; ALSO IN PROGRA
M TEST
   22.     COMPUTER SOC. PUBL. OFFICE LONG           METHODS, PART VIII, 1973, PP

   23.     BEACH, CALIF., 1974),                     277-290.
   24.     PP. 3.6.1-3.6.4.
   25.
   26.     LINDEN, T. A., "THE USE OF ABSTRACT
   27.     DATA TYPES TO SIMPLIFY PROGRAM
   28.     MODIFICATIONS," PROC. CONF. ON DATA:
   29.     ABSTRACTION DEFINITION AND STRUCTURE,
   30.     SALT LAKE CITY, UTAH, MAR. 22-24,
   31.     1976, ALSO IN SIGPLAN NOTICES 11,
   32.     SPEC.ISSUE, ASSOC. COMPUT.  MACH. -
   33.     FDT 8, NO. 2 (ASSOCIATION FOR
   34.     COMPUTING MACHINERY, N.Y.,N.Y.,
   35.     MAR. 1976), PP. 12-23
   36.
   37.     SADOWSKI, W. L. AND LOZIER, D. W.,
```

STEP 3
(Continued)

```
   38.     "A UNIFIED STANDARDS APPROCH TO
   39.     ALGORITHM TESTING," PROC. ACM
   40.     SIGPLAN SYMP. ON COMPUTER PROGRAM
   41.     TEST METHODS, CHAPEL HILL, N.C., JUNE
   42.     21-23, 1972; ALSO IN PROGRAM TEST
   43.     METHODS, PART VIII, 1973, PP.
   44.     277-290.
   45.
   46.     STILLMAN, R. B., "A SURVEY OF
   47.     TECHNIQUES FOR INCREASING SOFTWARE
   48.     RELIABILITY," PROC. SUMMER COMPUTER
   49.     SIMULATION CONF., MONTREAL, CANADA,
   50.     JULY 1973, PP. 1130-1133.
   51.
   52.     TESTING, VALIDATING, AND AUDITING
   53.
   54.     FONG, E., "A BENCHMARK TEST APPROACH FOR GENERALIZED
   55.     DATA BASE SOFTWARE," PROC. COMPCON FALL CONF. DIGEST,
   56.     WASHINGTON, D.C., SEPT. 8-10, 1975, PP. 7.
   57.
   58.     LYON, G., "SYNTAX-DIRECTED LEAST-ERRORS ANALYSIS
   59.     FOR CONTEXT-FREE LANGUAGES: A PRACTICAL APPROACH,"
   60.     COMMUN. ACM 17, NO. 1, JAN. 1974, PP. 3-14.
   61.
   62.     LYON, G. AND STILLMAN, R. B., "SIMPLE TRANSFORMS FOR
   63.     INSTRUMENTING FORTRAN DECKS," SOFTWARE PRACT. EXPER. 5,
   64.     NO. 4, OCT.-DEC. 1975, PP. 347-358.
   65.     $
COMMAND? ALIGN 54/64 LENGTH 37
COMMAND? L 54/64
   54.     FONG, E., "A BENCHMARK TEST APPROACH
   54.5    FOR GENERALIZED DATA BASE SOFTWARE,"
   55.     PROC. COMPCON FALL CONF. DIGEST,
   55.5    WASHINGTON, D.C., SEPT. 8-10, 1975,
   56.     PP. 7.
   56.5
```

STEP 4

STEP 4 (Continued)

```
   57.     LYON, G., "SYNTAX-DIRECTED
   57.5    LEAST-ERRORS ANALYSIS FOR
   59.     CONTEXT-FREE LANGUAGES: A PRACTICAL
   59.5    APPROACH," COMMUN. ACM 17, NO. 1,
   59.     JAN. 1974, PP. 3-14.
   59.5
   60.     LYON, G. AND STILLMAN, R. B.,
   60.5    "SIMPLE TRANSFORMS FOR INSTRUMENTING
   61.     FORTRAN DECKS," SOFTWARE PRACT.
   61.5    EXPER. 5, NO. 4, OCT.-DEC. 1975, PP.
   62.     347-359.
```

STEP 5

```
COMMAND? CH '-' TO '- ' IN 57.5
   57.5   LEAST- ERRORS ANALYSIS FOR
```

STEP 6

```
COMMAND? ALIGN 57/59 LENGTH 35
COMMAND? L 57/59
   57.     LYON, G., "SYNTAX-DIRECTED LEAST-
   57.5    ERRORS ANALYSIS FOR CONTEXT-FREE
   59.     LANGUAGES: A PRACTICAL APPROACH,"
   59.5    COMMUN. ACM 17, NO. 1, JAN. 1974,
   59.     PP. 3-14.
```

STEP 7

```
COMMAND? COPY 6/14 TO END
   74.     - LAST LINE.
```

STEP 8

```
COMMAND? CH 42/80 TO '' IN 6/23
    6.    DAVIS, R. M., "QUALITY SOFTWARE CAN

    7.    CHANGE THE COMPUTER INDUSTRY, "PROC.

    8.    OF THE SYMPOSIUM ON COMPUTER PROGRAM

    9.    TEST METHODS, CHAPEL HILL, N.C., JUNE

   10.    21-23, 1972; ALSO IN PROGRAM TEST

   11.    METHODS, W. C. HETZEL, ED.
```

STEP 8 (Continued)

```
12.     (PRENTICE-HALL, INC., ENGLEWOOD
13.     CLIFFS, N.J., 1973), PP.  303-311.
14.
15.     HUGHES, C. E.  AND WALKER, J. C.,
16.     "POPSS, A SYSTEM FOR MODELLING AND
17.     ANALYZING OPERATING SYSTEM RESOURCE
18.     ALLOCATION STRATEGIES," PROC. OF THE
19.     3RD TEXAS CONFERENCE ON COMPUTING
20.     SYSTEMS, AUSTIN, TEX., NOV. 7-8,
21.     1974, PAPER 74-CH0895-3C (IEEE
22.     COMPUTER SOC. PUBL. OFFICE LONG
23.     BEACH, CALIF., 1974),
COMMAND? CH 1/41 TO '' IN 66/74
66.      MILLER, E. F., JR. AND LINDAMOOD,
67.      G. E., STRUCTURED PROGRAMMING:
68.     TOP-DOWN APPROACH," DATAMATION 19,
69.     NO. 12, DEC.  1973 PP. 12-23
70.
71.     STEWART, S. L., "STAPLE, AN
72.     EXPERIMENTAL STRUCTURED PROGRAMMING
```

STEP 9

Appendix B

```
STEP 9         73.    LANGUAGE," COMPUT.  LANGUAGES 1, NO.
(Continued)    74.    1, JAN. 1975, PP. 61-71.

STEP 10      COMMAND? CH '-' TO '- ' IN 12
                12.    (PRENTICE- HALL, INC., ENGLEWOOD

             COMMAND? ALIGN 11/13 LENGTH 38
             COMMAND? CH ' (' TO '(' IN 11
                11.    METHODS, W. C. HETZEL, ED. (PRENTICE-

STEP 11      COMMAND? L 11/13
                11.     METHODS, W. C. HETZEL, ED. (PRENTICE-
                12.     HALL, INC., ENGLEWOOD CLIFFS, N.J.,
                13.     1973), PP.  303-311.
             COMMAND? L 6/23 66/L
                 6.     DAVIS, R. M., "QUALITY SOFTWARE CAN
                 7.     CHANGE THE COMPUTER INDUSTRY, "PROC.
                 8.     OF THE SYMPOSIUM ON COMPUTER PROGRAM
                 9.     TEST METHODS, CHAPEL HILL, N.C., JUNE
                10.     21-23, 1972; ALSO IN PROGRAM TEST
                11.     METHODS, W. C. HETZEL, ED. (PRENTICE-
                12.     HALL, INC., ENGLEWOOD CLIFFS, N.J.,
                13.     1973), PP.  303-311.
                14.
STEP 12         15.     HUGHES, C. E.  AND WALKER, J. C.,
                16.     "POPSS, A SYSTEM FOR MODELLING AND
                17.     ANALYZING OPERATING SYSTEM RESOURCE
                18.     ALLOCATION STRATEGIES," PROC. OF THE
                19.     3RD TEXAS CONFERENCE ON COMPUTING
                20.     SYSTEMS, AUSTIN, TEX., NOV. 7-8,
                21.     1974, PAPER 74-CH0895-3C (IEEE
                22.     COMPUTER SOC. PUBL. OFFICE LONG
                23.     BEACH, CALIF., 1974),
                66.      MILLER, E. F., JR. AND LINDAMOOD,
                67.      G. E., STRUCTURED PROGRAMMING:
                68.     TOP-DOWN APPROACH," DATAMATION 19,
                69.     NO. 12, DEC.  1973 PP. 12-23
```

STEP 12
(Continued)

```
   70.
   71.      STEWART, S. L., "STAPLE, AN
   72.      EXPERIMENTAL STRUCTURED PROGRAMMING
   73.      LANGUAGE," COMPUT. LANGUAGES 1, NO.
   74.      1, JAN. 1975, PP. 61-71.
COMMAND? CH '"' TO '""' IN 67
   67.      G. E., "STRUCTURED PROGRAMMING!

COMMAND? CH 1/1 TO '' IN 66 67
   66.     MILLER, E. F., JR. AND LINDAMOOD,

   67.     G. E., "STRUCTURED PROGRAMMING!

COMMAND? L 6/23 66/L
    6.      DAVIS, R. M., ""QUALITY SOFTWARE CAN
    7.      CHANGE THE COMPUTER INDUSTRY, "PROC.
    8.      OF THE SYMPOSIUM ON COMPUTER PROGRAM
    9.      TEST METHODS, CHAPEL HILL, N.C., JUNE
   10.      21-23, 1972; ALSO IN PROGRAM TEST
   11.      METHODS, W. C. HETZEL, ED. (PRENTICE-
   12.      HALL, INC., ENGLEWOOD CLIFFS, N.J.,
   13.      1973), PP.  303-311.
   14.
   15.      HUGHES, C. E.  AND WALKER, J. C.,
   16.      "POPSS, A SYSTEM FOR MODELLING AND
   17.      ANALYZING OPERATING SYSTEM RESOURCE
   18.      ALLOCATION STRATEGIES," PROC. OF THE
   19.      3RD TEXAS CONFERENCE ON COMPUTING
   20.      SYSTEMS, AUSTIN, TEX., NOV. 7-8,
   21.      1974, PAPER 74-CH0895-3C (IEEE
```

```
STEP 12          22.     COMPUTER SOC. PUBL. OFFICE LONG
(Continued)      23.     BEACH, CALIF., 1974),
                 66.     MILLER, E. F., JR. AND LINDAMOOD,
                 67.     G. E., "STRUCTURED PROGRAMMING:
                 68.     TOP-DOWN APPROACH," DATAMATION 19,
                 69.     NO. 12, DEC.  1973 PP. 12-23
                 70.
                 71.     STEWART, S. L., "STAPLE, AN
                 72.     EXPERIMENTAL STRUCTURED PROGRAMMING
                 73.     LANGUAGE," COMPUT.  LANGUAGES 1, NO.
                 74.     1, JAN. 1975, PP. 61-71.
STEP 13        COMMAND? MOVE 66/74 TO 36.1
                 36.9    - LAST LINE.
STEP 14        COMMAND? ALIGN 6/36.9 INDENT 1 EVEN LENGTH 39
STEP 15        COMMAND? MOVE 37/50 TO END
                 79.     - LAST LINE.
STEP 16        COMMAND? ALIGN 54/62 INDENT 1 EVEN LENGTH 39
STEP 17        COMMAND? L
                  1.     $
                  2.     SOFTWARE ENGINEERING (SEE ALSO
                  3.     PROGRAMMING LANGUAGES AND TESTING,
                  4.     VALIDATING AND AUDITING)
                  5.
                  6.      DAVIS, R. M., "QUALITY SOFTWARE CAN
                  6.5     CHANGE THE COMPUTER INDUSTRY, "PROC.
                  7.      OF THE SYMPOSIUM ON COMPUTER PROGRAM
                  7.5     TEST METHODS, CHAPEL HILL, N.C., JUNE
                  8.      21-23, 1972;  ALSO IN PROGRAM TEST
                  8.5     METHODS, W. C. HETZEL, ED. (PRENTICE-
                  9.      HALL, INC., ENGLEWOOD CLIFFS, N.J.,
                  9.5     1973), PP.  303-311.
                 10.
                 10.5     HUGHES, C. E.  AND WALKER, J. C.,
                 11.      "POPSS, A SYSTEM FOR MODELLING AND
                 11.5     ANALYZING OPERATING SYSTEM RESOURCE
```

STEP 17
(Continued)

ALLOCATION STRATEGIES," PROC. OF THE 3RD TEXAS CONFERENCE ON COMPUTING SYSTEMS, AUSTIN, TEX., NOV. 7-9, 1974, PAPER 74-CH0895-3C (IEEE COMPUTER SOC. PUBL. OFFICE LONG BEACH, CALIF., 1974), PP. 3.6.1-3.6.4.

LINDEN, T. A., "THE USE OF ABSTRACT DATA TYPES TO SIMPLIFY PROGRAM MODIFICATIONS," PROC. CONF. ON DATA: ABSTRACTION DEFINITION AND STRUCTURE, SALT LAKE CITY, UTAH, MAR. 22-24, 1976, ALSO IN SIGPLAN NOTICES 11, SPEC.ISSUE, ASSOC. COMPUT. MACH. - FDT 8, NO. 2 (ASSOCIATION FOR COMPUTING MACHINERY, N.Y.,N.Y., MAR. 1976), PP. 12-23

MILLER, E. F., JR. AND LINDAMOOD, G. E., "STRUCTURED PROGRAMMING: TOP-DOWN APPROACH," DATAMATION 19, NO. 12, DEC. 1973 PP. 12-23

STEWART, S. L., "STAPLE, AN EXPERIMENTAL STRUCTURED PROGRAMMING LANGUAGE," COMPUT. LANGUAGES 1, NO. 1, JAN. 1975, PP. 61-71.

TESTING, VALIDATING, AND AUDITING

FONG, E., "A BENCHMARK TEST APPROACH FOR GENERALIZED DATA BASE SOFTWARE," PROC. COMPCON FALL CONF. DIGEST, WASHINGTON, D.C., SEPT. 9-10, 1975, PP. 7.

STEP 17
(Continued)

```
57.      LYON, G., "SYNTAX-DIRECTED LEAST-
57.5     ERRORS ANALYSIS FOR CONTEXT-FREE
58.      LANGUAGES: A PRACTICAL APPROACH,"
58.5     COMMUN. ACM 17, NO. 1, JAN. 1974, PP.
59.      3-14.
59.5
60.      LYON, G. AND STILLMAN, R. B.,
60.5     "SIMPLE TRANSFORMS FOR INSTRUMENTING
61.      FORTRAN DECKS," SOFTWARE PRACT.
61.5     EXPER. 5, NO. 4, OCT.-DEC. 1975, PP.
62.      347-358.
65.     $
66.     SADOWSKI, W. L. AND LOZIER, D. W.,
67.     "A UNIFIED STANDARDS APPROCH TO
68.     ALGORITHM TESTING," PROC. ACM
69.     SIGPLAN SYMP. ON COMPUTER PROGRAM
70.     TEST METHODS, CHAPEL HILL, N.C., JUNE
71.     21-23, 1972;  ALSO IN PROGRAM TEST
72.     METHODS, PART VIII, 1973, PP.
73.     277-290.
74.
75.     STILLMAN, R. B., "A SURVEY OF
76.     TECHNIQUES FOR INCREASING SOFTWARE
77.     RELIABILITY," PROC. SUMMER COMPUTER
78.     SIMULATION CONF., MONTREAL, CANADA,
79.     JULY 1973, PP. 1130-1133.
```

STEP 18

```
COMMAND? CH 'PP.' TO '' IN 14.5 58.5 61.5 72
   14.5   BEACH, CALIF., 1974),

   58.5   COMMUN. ACM 17, NO. 1, JAN. 1974,

   61.5   EXPER. 5, NO. 4, OCT.-DEC. 1975,

   72.   METHODS, PART VIII, 1973,

COMMAND? CH 2 IN 15 59 62 TO 'PP.
   15.    PP.  3.6.1-3.6.4.

   59.    PP.  3-14.

   62.    PP.  347-358.
```

```
STEP 18      COMMAND? CH 1 TO 'PP.  ' IN 73
(Continued)     73.     PP.  277-290.

STEP 19      COMMAND? L 'SADOWSKI'
                66.     SADOWSKI, W. L. AND LOZIER, D. W.,
             COMMAND? L 66/L
                66.     SADOWSKI, W. L. AND LOZIER, D. W.,
                67.     "A UNIFIED STANDARDS APPROCH TO
                68.     ALGORITHM TESTING," PROC. ACM
                69.     SIGPLAN SYMP. ON COMPUTER PROGRAM
                70.     TEST METHODS, CHAPEL HILL, N.C., JUNE
STEP 20         71.     21-23, 1972; ALSO IN PROGRAM TEST
                72.     METHODS, PART VIII, 1973,
                73.     PP.  277-290.
                74.
                75.     STILLMAN, R. B., "A SURVEY OF
                76.     TECHNIQUES FOR INCREASING SOFTWARE
                77.     RELIABILITY," PROC. SUMMER COMPUTER
                78.     SIMULATION CONF., MONTREAL, CANADA,
                79.     JULY 1973, PP. 1130-1133.
             COMMAND? CH 42/80 IN 2/10.5 USING 66/79
                 2.    SOFTWARE ENGINEERING (SEE ALSO         SADOWSKI, W. L. AND LOZIER, D
             . W.,

                 3.    PROGRAMMING LANGUAGES AND TESTING,     "A UNIFIED STANDARDS APPROCH
              TO
STEP 21
                 4.    VALIDATING AND AUDITING)               ALGORITHM TESTING," PROC. AC
             M

                 5.                                           SIGPLAN SYMP. ON COMPUTER PRO
             GRAM

                 6.     DAVIS, R. M., "QUALITY SOFTWARE CAN   TEST METHODS, CHAPEL HILL, N.
             C., JUNE
```

STEP 21 (Continued)

```
     6.5     CHANGE THE COMPUTER INDUSTRY, "PROC.   21-23, 1972; ALSO IN PROGRAM
TEST
     7.      OF THE SYMPOSIUM ON COMPUTER PROGRAM    METHODS, PART VIII, 1973,
     7.5     TEST METHODS, CHAPEL HILL, N.C., JUNE   PP.  277-290.
     8.      21-23, 1972; ALSO IN PROGRAM TEST
     8.5     METHODS, W. C. HETZEL, ED. (PRENTICE-   STILLMAN, R. B., "A SURVEY O
F
     9.      HALL, INC., ENGLEWOOD CLIFFS, N.J.,     TECHNIQUES FOR INCREASING SOF
TWARE
     9.5     1973), PP.  303-311.                    RELIABILITY," PROC. SUMMER C
OMPUTER
    10.                                              SIMULATION CONF., MONTREAL, C
ANADA,
    10.5     HUGHES, C. E.  AND WALKER, J. C.,       JULY 1973, PP. 1130-1133.
COMMAND? DELETE 66/79
COMMAND? L 1/11
     1.      $
     2.      SOFTWARE ENGINEERING (SEE ALSO           SADOWSKI, W. L. AND LOZIER,
D. W.,
     3.      PROGRAMMING LANGUAGES AND TESTING,       "A UNIFIED STANDARDS APPROC
H TO
     4.      VALIDATING AND AUDITING)                 ALGORITHM TESTING," PROC. A
CM
     5.                                               SIGPLAN SYMP. ON COMPUTER PR
OGRAM
     6.       DAVIS, R. M., "QUALITY SOFTWARE CAN    TEST METHODS, CHAPEL HILL, N
.C., JUNE
     6.5      CHANGE THE COMPUTER INDUSTRY, "PROC.   21-23, 1972; ALSO IN PROGRA
M TEST
     7.       OF THE SYMPOSIUM ON COMPUTER PROGRAM    METHODS, PART VIII, 1973,
```

STEP 22

STEP 22
(Continued)

```
   7.5     TEST METHODS, CHAPEL HILL, N.C., JUNE   PP.  277-290.
   9.      21-23, 1972;  ALSO IN PROGRAM TEST
   9.5     METHODS, W. C. HETZEL, ED. (PRENTICE-   STILLMAN, R. B., "A SURVEY
OF
   9.      HALL, INC., EN' EWOOD CLIFFS, N.J.,     TECHNIQUES FOR INCREASING SO
FTWARE
   9.5     1973), PP.  303-311.                    RELIABILITY," PROC. SUMMER
COMPUTER
  10.                                              SIMULATION CONF., MONTREAL,
CANADA,
  10.5     HUGHES, C. E.  AND WALKER, J. C.,       JULY 1973, PP. 1130-1133.
  11.      "POPSS, A SYSTEM FOR MODELLING AND
COMMAND? CH 40 TO ' ' IN 6/6.5
   6.      DAVIS, R. M., "QUALITY SOFTWARE CAN    TEST METHODS, CHAPEL HILL, N
.C., JUNE

   6.5     CHANGE THE COMPUTER INDUSTRY, "PROC.   21-23, 1972;  ALSO IN PROGRA
M TEST

COMMAND? L 1/11
   1.      $
   2.      SOFTWARE ENGINEERING (SEE ALSO          SADOWSKI, W. L. AND LOZIER,
D. W.,
   3.      PROGRAMMING LANGUAGES AND TESTING,      "A UNIFIED STANDARDS APPROC
H TO
   4.      VALIDATING AND AUDITING)                ALGORITHM TESTING," PROC. A
CM
   5.                                              SIGPLAN SYMP. ON COMPUTER PR
OGRAM
   6.       DAVIS, R. M., "QUALITY SOFTWARE CAN    TEST METHODS, CHAPEL HILL,
N.C., JUNE
   6.5      CHANGE THE COMPUTER INDUSTRY, "PROC.   21-23, 1972;  ALSO IN PROGR
AM TEST
   7.       OF THE SYMPOSIUM ON COMPUTER PROGRAM   METHODS, PART VIII, 1973,
   7.5      TEST METHODS, CHAPEL HILL, N.C., JUNE  PP.  277-290.
   9.       21-23, 1972;  ALSO IN PROGRAM TEST
```

AD-A082 209 GENERAL SERVICES ADMINISTRATION WASHINGTON DC AUTOMA--ETC F/G 9/2
USE AND SPECIFICATIONS OF REMOTE TERMINAL EMULATION IN ADP SYST--ETC(U)
AUG 79
UNCLASSIFIED FPR-1-4.11 NL

3 OF 3
AD
A082209

END
DATE
FILMED
4-80
DTIC

STEP 22 (Continued)

```
    9.5     METHODS, W. C. HETZEL, ED. (PRENTICE-    STILLMAN, R. B., "A SURVEY
OF
    9.      HALL, INC., ENGLEWOOD CLIFFS, N.J.,      TECHNIQUES FOR INCREASING SO
FTWARE
    9.5     1973), PP.  303-311.                     RELIABILITY," PROC. SUMMER
COMPUTER
   10.                                               SIMULATION CONF., MONTREAL,
CANADA,
   10.5     HUGHES, C. E.  AND WALKER, J. C.,        JULY 1973, PP. 1130-1133.
   11.      'POPSS, A SYSTEM FOR MODELLING AND
COMMAND? L UNN MARKER $
```

STEP 23

SOFTWARE ENGINEERING (SEE ALSO PROGRAMMING LANGUAGES AND TESTING, VALIDATING AND AUDITING)

DAVIS, R. M., "QUALITY SOFTWARE CAN CHANGE THE COMPUTER INDUSTRY, "PROC. OF THE SYMPOSIUM ON COMPUTER PROGRAM TEST METHODS, CHAPEL HILL, N.C., JUNE 21-23, 1972; ALSO IN PROGRAM TEST METHODS, W. C. HETZEL, ED. (PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N.J., 1973), PP. 303-311.

HUGHES, C. E. AND WALKER, J. C., "POPSS, A SYSTEM FOR MODELLING AND

SADOWSKI, W. L. AND LOZIER, D. W., "A UNIFIED STANDARDS APPROCH TO ALGORITHM TESTING," PROC. ACM SIGPLAN SYMP. ON COMPUTER PROGRAM TEST METHODS, CHAPEL HILL, N.C., JUNE 21-23, 1972; ALSO IN PROGRAM TEST METHODS, PART VIII, 1973, PP. 277-290.

STILLMAN, R. B., "A SURVEY OF TECHNIQUES FOR INCREASING SOFTWARE RELIABILITY," PROC. SUMMER COMPUTER SIMULATION CONF., MONTREAL, CANADA, JULY 1973, PP. 1130-1133.

STEP 23
(Continued)

ANALYZING OPERATING SYSTEM RESOURCE ALLOCATION STRATEGIES," PROC. OF THE 3RD TEXAS CONFERENCE ON COMPUTING SYSTEMS, AUSTIN, TEX., NOV. 7-9, 1974, PAPER 74-CH0995-3C (IEEE COMPUTER SOC. PUBL. OFFICE LONG BEACH, CALIF., 1974), PP. 3.6.1-3.6.4.

LINDEN, T. A., "THE USE OF ABSTRACT DATA TYPES TO SIMPLIFY PROGRAM MODIFICATIONS," PROC. CONF. ON DATA: ABSTRACTION DEFINITION AND STRUCTURE, SALT LAKE CITY, UTAH, MAR. 22-24, 1976, ALSO IN SIGPLAN NOTICES 11, SPEC.ISSUE, ASSOC. COMPUT. MACH. - FDT 9, NO. 2 (ASSOCIATION FOR COMPUTING MACHINERY, N.Y.,N.Y., MAR. 1976), PP. 12-23

MILLER, E. F., JR. AND LINDAMOOD, G. E., "STRUCTURED PROGRAMMING: TOP-DOWN APPROACH," DATAMATION 19, NO. 12, DEC. 1973 PP. 12-23

STEWART, S. L., "STAPLE, AN EXPERIMENTAL STRUCTURED PROGRAMMING LANGUAGE," COMPUT. LANGUAGES 1, NO. 1, JAN. 1975, PP. 61-71.

TESTING, VALIDATING, AND AUDITING

FONG, E., "A BENCHMARK TEST APPROACH FOR GENERALIZED DATA BASE SOFTWARE," PROC. COMPCON FALL CONF. DIGEST, WASHINGTON, D.C., SEPT. 9-10, 1975, PP. 7.

LYON, G., "SYNTAX-DIRECTED LEAST-

STEP 23 (Continued)

```
ERRORS ANALYSIS FOR CONTEXT-FREE
LANGUAGES: A PRACTICAL APPROACH,"
COMMUN. ACM 17, NO. 1, JAN. 1974,
PP.  3-14.

LYON, G. AND STILLMAN, R. B.,
"SIMPLE TRANSFORMS FOR INSTRUMENTING
FORTRAN DECKS," SOFTWARE PRACT.
EXPER. 5, NO. 4, OCT.-DEC. 1975,
PP.  347-359.
```

STEP 24

```
COMMAND? NUM
   66.    - LAST LINE.
```

STEP 25

```
COMMAND? SAVE E1001 ON USER02
FE93.E1001 SAVED ON USER02
```

STEP 26

```
COMMAND? LOGOFF
OK TO CLEAR? YES
READY
LOGOFF


FE93 LOGGED OFF
```

GLOSSARY

This glossary supplements FIPS PUB 11-1, "Vocabulary for Information Processing," and should be used during benchmark tests using remote terminal emulation. A number of sources have been consulted; the definitions given here are as consistent with everyday usage as possible.

ADP SERVICE PROCUREMENT--an acquisition that results in the Government obtaining the use of ADP equipment (ADPE) containing at least one general purpose central processing unit that is either owned and operated or leased and operated by a contractor; the ADPE may be either dedicated for the exclusive use of the acquiring Government organization or shared by many Government and/or non-government organizations

ADP SYSTEM PROCUREMENT--an acquisition that results in the lease and/or purchase by the Government of ADP equipment (ADPE) containing at least one general purpose central processing unit; the acquired ADPE may be operated by either government or contractor personnel

APPLICATION I/O PAIR--an I/O pair that is explicitly related to accomplishing a user function from a remote TP device; the nature and number required to accomplish a single user function vary from computer system to computer system

ASCII--abbreviation for American Standard Code for Information Interchange (See FIPS PUB 1.)

BENCHMARK DISCREPANCY--a difference between a technical or procedural aspect of a benchmark test, as conducted by a vendor, and the manner that the procuring agency intended for that aspect to be accomplished

BENCHMARK MIX--the collection of user workload elements (e.g., data files, batch jobs, interactive commands) that comprises the test workload of a benchmark test and that typifies the processing environment under evaluation; synonymous with test workload

BENCHMARK MIX DEMONSTRATION--see live test demonstration

BENCHMARK MIX EXECUTION--a single execution of a specific benchmark mix on a specific SUT

BENCHMARK REPEATABILITY--the degree of similarity between two executions of the same benchmark mix on the same SUT

BENCHMARK REPRESENTATIVENESS--the degree to which a benchmark test duplicates an operational processing environment anticipated to occur during the contractual life of an acquisition

BENCHMARK TEST--the specific collection of elements used to determine selected execution characteristics of a SUT; example elements include a benchmark mix and execution procedures

BENCHMARK UNIFORMITY--the degree of similarity between the test workload demands imposed on different SUT's by the execution of the same benchmark mix

BENCHMARK VERIFICATION--the act of determining the degree to which a vendor conducted a benchmark test in the manner intended by the procuring agency

BENCHMARKING--the process of experimentally imposing a test workload on one or more ADP system components to determine selected execution characteristics of the components

CAPABILITY DEMONSTRATION--see functional test

CAPACITY TEST--a benchmark test used to determine if a SUT can accomplish a specific, often large, set of user work items at a required level of performance; sometimes referred to as a load test or a timed test

CONFIGURATION CONSTRAINTS--restrictions imposed by a procuring agency on the number, types, characteristics, and/or installation schedules of hardware and software components bid by vendors

COST RISK--the likelihood that an agency will pay more for the ADP configuration(s) ultimately acquired than is necessary to satisfy the agency's ADP requirements during the contractual life of the acquisition

DATA COMMUNICATION INPUT-OUTPUT PAIR--an exchange of functionally related data transmissions by a TP device and the SUT

DIALOGUE--the actions and inputs of the operator of a teleprocessing device that are required to accomplish one or more user functions

DRIVER--remote terminal emulation component, external to the SUT, which introduces specified TP workload demands to the ADP system being tested

EMULATION BENCHMARK TEST--a benchmark test that uses remote terminal emulation

FUNCTIONAL DEMONSTRATION--synonymous with functional test

FUNCTIONAL TEST--a benchmark test used to determine if a SUT can accomplish a specific user work item without regard to completion time and other workload demands; synonymous with functional demonstration and capability demonstration

INTERMEDIATE DEVICE--a teleprocessing device used to connect one or more remote devices to a computer system; configured between a computer system and one or more remote devices

I/O PAIR--abbreviation for a data communication input-output pair

LIVE TEST DEMONSTRATION--the period of time during which a user requires a vendor to perform certain user-witnessed activities necessary to complete one or more benchmark tests; abbreviated LTD

LOAD TEST--see capacity test

LTD--abbreviation for Live Test Demonstration

MISSION RISK--the likelihood that the ADP configuration(s) ultimately acquired will fail to satisfy an agency's mission requirements at any point during the contractual life of the acquisition

MIX--see benchmark mix

MONITOR--a remote terminal emulation component, external to the SUT, which records data descriptive of the RTE/SUT interaction; may or may not be an integral component of an RTE

REMOTE DEVICE--a teleprocessing device where user work items originate

REMOTE TERMINAL EMULATION--a benchmarking technique in which a "driver" computer system external to, and independent of, the SUT (1) connects to the SUT through the SUT's communication device interfaces, and (2) interacts with the SUT as if the driver were a set of teleprocessing devices and operators; integral to this technique is a monitor external to the SUT which captures data descriptive of the driver/SUT interaction

REMOTE TERMINAL EMULATOR--a specific implementation of a teleprocessing workload driver; integral to it may or may not be a monitor; a necessary aspect of remote terminal emulation; abbreviated RTE

REPEATABILITY--see benchmark repeatability

REPRESENTATIVENESS--see benchmark representativeness

REQUIREMENT SPECIFICATIONS--the description of the user workload demands that the system(s) ultimately acquired must be able to complete and the acceptable level(s) of performance for completing these demands; synonymous with workload specifications

RESPONSE TIME--the elapsed time from the last keystroke of an operator input at an interactive device until the first printable character of the resulting SUT response appears at the device

RTE--abbreviation for remote terminal emulator

SCENARIO--a system- and vendor-independent description of a group of user TP workload demands to be performed during a benchmark mix execution; expressed as user functions

SCENE--the dynamic interaction of an RTE and a SUT during a benchmark mix execution

SCENE MONITORING--the recording of data descriptive of a scene

SCRIPT--the set of instructions, data, and procedures that causes a particular RTE to impose specific TP workload demands on a given SUT; includes both commands to control the RTE and the dialogue; partially based on user functions defined in a scenario

SERVICE PROCUREMENT--See ADP service procurement

STIMULATION--the use of an RTE to impose TP workload demands on a SUT

SUT--abbreviation for system under test

SYSTEM PROCUREMENT--See ADP system procurement

SYSTEM SIZING--the process of determining a configuration of ADP hardware and software components that can accomplish a specific set of workload demands at a required level of performance

SYSTEM UNDER TEST--the collection of ADP components whose performance characteristics are determined by a benchmark test; abbreviated SUT

TELEPROCESSING--a form of information handling in which a data processing system utilizes data communication facilities

TEST WORKLOAD--synonymous with benchmark mix

THROUGHPUT--the number of user work items successfully completed within a predefined time interval

TIMED TEST--see capacity test

TP--abbreviation for teleprocessing

TRANSMISSION BLOCK--a group of digits transmitted as a unit, over which a coding procedure is usually applied for synchronization or error control purposes

TURNAROUND TIME--the time interval between the initiation of a user work item and the successful completion of the work item

UNIFORMITY--see benchmark uniformity

USER FUNCTION--an action that must be performed to satisfy an organization's data processing requirements; a single, logically-related action included in a benchmark mix; synonymous with user work item

USER WORK ITEM--synonymous with user function

VERIFICATION--see benchmark verification

WORKLOAD DEMAND--some number of user functions

WORKLOAD MIX--see benchmark mix

WORKLOAD SPECIFICATIONS--synonymous with requirement specifications

BIBLIOGRAPHY

DATA COMMUNICATION AND INFORMATION PROCESSING STANDARDS AND GUIDELINES[1]

American National Standards Institute (ANSI). "Advanced Data Communication Control Procedures (ADCCP)," ANSI Standard X3.66-1979. New York, NY: ANSI, January 1979.

_____. "Code for Information Interchange," ANSI X3.4-1977. New York, NY: ANSI, 1977.

_____. "Distributed Systems Reference Model," Draft 4. New York, NY: ANSI, February 1978.

_____. "Magnetic Tape Labels and File Structure for Information Interchange," ANSI X3.27-1977. New York, NY: ANSI, 1977.

_____. "Procedures for the Use of the Communications Control Character of American National Standard Code for Information Interchange in Specified Data Communication Links," ANSI X3.28-1976. New York, NY: ANSI, 1976.

_____. "Recorded Magnetic Tape for Information Interchange (1600 CPI, Phase Encoded)," ANSI X3.39-1973. New York, NY: ANSI, 1975.

_____. "Recorded Magnetic Tape for Information Interchange (6250 CPI, Group Coded Recording)," ANSI X3.54-1976. New York, NY: ANSI, 1976

_____. "Representations of Local Time of the Day for Information Interchange," ANSI X3.43-1977. New York, NY: ANSI, 1977.

Electronic Industries Association (EIA). "General Purpose 37-Position and 9-Position Interface for Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) Employing Serial Binary Data Interchange," EIA RS-449. New York, NY: November 1977.

---

[1]Contact the General Services Administration, the National Bureau of Standards, and the National Communication System to obtain a complete listing of the latest standards and guidelines.

_____. "Interface between Data terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange," EIA RS-232-C. New York, NY: August 1969.

International Telecommunications Union. "Interface for Terminals Operating in the Packet Mode on Public Data Networks," CCITT X.25. Geneva, SW: International Telecommunications Union, April 1977.

National Bureau of Standards (NBS). "Bit Sequencing of the Code for Information Interchange in Serial-By-Bit Data Transmission," FIPS PUB 16-1. Washington, DC: NBS, September 1977.

_____. "Calendar Date," FIPS PUB 4. Washington, DC: NBS, November 1968.

_____. "Character Structure and Character Parity Sense for Serial-By-Bit Data Communications in the Code for Information Interchange," FIPS PUB 17-1. Washington, DC: NBS, September 1977.

_____. "Code Extension Techniques in 7 or 8 Bits," FIPS PUB 35. Washington, DC: NBS, June 1975.

_____. "Code for Information Interchange," FIPS PUB 1. Washington, DC: NBS, November 1968.

_____. "Data Encryption Standard," FIPS PUB 46. Washington, DC: NBS, January, 1977.

_____. "Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment," FIPS PUB 42-1. Washington, DC: NBS, May 1977.

_____. "Guidelines for the Measurement of Interactive Computer Service Turnaround Time and Response Time," FIPS PUB 57. Washington, DC: NBS, August 1978.

_____. "Vocabulary for Information Processing," FIPS PUB 11-1. Washington, DC: NBS, September 1977.

National Communication System. "Bit Oriented Data Link Control Procedures," Federal Standard 1003. Washington, DC: National Communication System, 1979.

FEDERAL ADP ACQUISITION POLICY AND REGULATIONS[2]

General Services Administration (GSA). "ADP and Telecommunications Management Policy," Federal Property Management Regulations, Title 41, Code of Federal Regulations, Chapter 101, Part 101-35. Washington, DC: GSA, June 1978.

_____. "ADP Management," Federal Property Management Regulations, Title 41, Code of Federal Regulations, Chapter 101, Part 101-36. Washington, DC: GSA, June 1978.

_____. "Federal Procurement Regulations: Amendment 181," Federal Procurement Regulations, Title 41, Chapter 1, Subpart 1-4.11. Washington, DC: GSA, August 1977.

_____. "General Instructions to Offerors Governing Proposal Preparation," in Standard Solicitation Documents. Washington, DC: GSA/ADTS (continuously updated).

_____. "Guidance to Federal Agencies on the Preparation of Specifications, Selection, and Acquisition of Automatic Data Processing Systems," in Standard Solicitation Documents. Washington, DC: GSA/ADTS (continuously updated).

_____. "Major System Acquisitions for Automatic Data Processing (ADP) and Telecommunications," Federal Procurement Regulations, Title 41, Code of Federal Regulations, Chapter 1, Temporary Regulation 47. Washington, DC: GSA, September 1978.

_____. "Procurement and Contracting for Government-Wide Automated Data Processing Equipment, Software, Maintenance, and Supplies," Federal Procurement Regulations, Title 41, Code of Federal Regulations, Chapter 1, Subpart 1-4.11. Washington, DC: GSA, September 1976.

_____. "Solicitation Document for ADP Systems," in Standard Solicitation Documents. Washington, DC: GSA/ADTS (continuously updated).

---

[2]Contact the Office of Management and Budget and the General Services Administration to obtain a complete listing of the latest policies and regulations.

_____. "Telecommunications Management," Federal Property Management Regulations, Title 41, Code of Federal Regulations, Chapter 101, Part 101-37. Washington, DC: GSA, June 1978.

_____. "Use of Benchmarks and Remote Terminal Emulation for Performance Validation in the Procurement of Automated Data Processing (ADP) Systems and Services," Federal Procurement Regulations, Title 41, Code of Federal Regulations, Chapter 1, Temporary Regulation 49 and Supplement 1. Washington, DC: GSA, August 1979.

_____. "Use of Small Purchase Procedures and Schedule Contracts for Automatic Data Processing (ADP) Requirements," Federal Procurement Regulations, Title 41, Code of Federal Regulations, Chapter 1, Temporary Regulation 46. Washington, DC: GSA, September 1978.

Office of Management and Budget (OMB). "Cost Comparison Handbook," Supplement No. 1, Circular A-76. Washington, DC: OMB, March 1979.

_____. "Major System Acquisitions," Circular A-109. Washington, DC: OMB, April 1976.

_____. "Policies for Acquiring Commercial or Industrial Products and Services Needed by the Government," Circular A-76 (Revised). Washington, DC: OMB, March 29, 1979.

BENCHMARKING DURING ADP SYSTEM ACQUISITION

Abrams, M. D., and Hayden, H.P. "Application of a Network Monitor to the Selection of a Time Shared Computing System," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publication 500-41. Washington, DC: National Bureau of Standards, October 1978, pp. 15-25.

Benwell, N. "Benchmarking: Computer Evaluation and Measurement." Washington, DC: Hemisphere, October 1974.

Buchanan, I., and Duce, D.A. "An Interactive Benchmark for a Multi-User Minicomputer System." Performance Evaluation Review, Fall 1976, pp. 5-17.

Davies, D. J. M. "Benchmarking in Selection of Timesharing System," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publication 500-41. Washington, DC: National Bureau of Standards, October 1978, pp. 27-36.

Gilbert, D. M., et al. "Guidance for Sizing ADP Systems (ADPS'S)," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publication 500-41. Washington, DC: National Bureau of Standards, October 1978, pp. 305-330.

Mamrak, S. A., and Amer, P. D. "A Methodology for the Selection of Interactive Computer Services," Special Publication 500-44. Washington, DC: National Bureau of Standards, January 1979.

Mukherjee, A., and Lacro, J. "An Improved Benchmark Performance Evaluation Technique for Vendor Selection Studies," in 1976 Computer Performance Evaluation Users Group Conference. Washington, DC: National Bureau of Standards, November 1976, pp. 197-201.

National Bureau of Standards (NBS). "Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment," FIPS 42-1. Washington, DC: NBS, May 1977.

Shetler, A. C., and Bell, T. E. "Computer Performance Analysis: Controlled Testing," R-1436-DCA. Santa Monica, CA: The Rand Corporation, April 1974.

Shetler, A. C. "Controlled Testing for Computer Performance Evaluation," in 1974 National Computer Conference. Montvale, NJ: AFIPS Conference Proceedings, May 1974, pp. 693-699.

Timmreck, E. M. "Computer Selection Methodology." Computing Surveys, V5, N4, December 1973, pp. 199-222.

Walkowicz, J. L. "Benchmarking and Workload Definition: A Selected Bibliography with Abstracts." Washington, DC: Government Printing Office, November 1974.

Waters, R. E. "Selection of ADPS for the Air Force Academy: A Case Study," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publication 500-18. Washington, DC: National Bureau of Standards, October 1977, pp. 71-74.

Wyrick, T. F. "Benchmarking Distributed Systems: Objectives and Techniques," in Performance of Computer Installations, ed. D. Ferrari. New York, NY: North-Holland, 1978.

REMOTE TERMINAL EMULATION

Arthur, C. T. "Remote Terminal Emulator Development and Application Criteria," in 1977 National Computer Conference. Montvale, NJ: AFIPS Conference Proceedings, June 1977, pp. 733-739.

General Services Administration (GSA). "Remote Terminal Emulation Specifications for Federal ADP System Procurements (Draft)," Report CDD 79-1. Washington, DC: GSA/ADTS, October 1978.

Hyman, B. "Stability and Workload Definition for Time Sharing Systems," TG-13 Meeting Minutes. Washington, DC: FIPS Coordinating and Advisory Committee, July 1975.

Tendolkar, N. N. "Determination of Non-Steady State Conditions in Performance Measurement Runs," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publication 500-18. Washington, DC: National Bureau of Standards, September 1977, pp. 87-94.

Tobagi, F. A., et al. "Modeling and Measurement Techniques in Packet Communication Networks." Proceedings of the IEEE, V66, N11, November 1978, pp. 1423-1447.

Trehan, V. "Problems in Remote Terminal Emulation," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publicaton 500-41. Washington, DC: National Bureau of Standards, October 1978, pp. 37-61.

Watkins, S. W., and Abrams, M. D. "Remote Terminal Emulation in the Procurement of Teleprocessing Systems," in 1977 National Computer Conference. Montvale, NJ: AFIPS Conference Proceedings, June 1977, pp. 723-727.

_____. "Survey of Remote Terminal Emulators," Special Publication 500-4. Washington, DC: National Bureau of Standards, April 1977.

Wyrick, T. F., and Findley, G. W. "Incorporating Remote Terminal Emulation into the Federal ADP Procurement Process," in Computer Performance Evaluation Users Group Conference Proceedings, Special Publication 500-41. Washington, DC: National Bureau of Standards, October 1978, pp. 5-14.

Wyrick, T. F. "Procurement Validation Case Studies: Concepts and Issues Relevant to the Use of Remote Terminal Emulation in Teleprocessing Procurements," Report CS77-6. Washington, DC: GSA/ADTS, May 1977.

*U.S. GOVERNMENT PRINTING OFFICE: 1980 620-032/790 1-3